UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

Natural Language Processing and Machine Learning Techniques to Solve a Breast Cancer Clinical Trial ECOG-Classification Problem

Author:
Pablo Eliseo Reynoso Aguirre

Supervisor:
Dr. Horacio Rodríguez Hontoria
Dr. Lluís Antoni Belanche Muñoz

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Artificial Intelligence
Natural Language Processing & Machine Learning Research

April 16, 2018

Declaration of Authorship

I, Pablo Eliseo Reynoso Aguirre, declare that this thesis titled, "Natural Language Processing and Machine Learning Techniques to Solve a Breast Cancer Clinical Trial ECOG-Classification Problem" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:
_____ Date:
_____

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

Alan Turing

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Abstract

Facultat d'Informàtica de Barcelona
Natural Language Processing & Machine Learning Research

Master of Science

Natural Language Processing and Machine Learning Techniques to Solve a Breast Cancer Clinical Trial ECOG-Classification Problem

by Pablo Eliseo Reynoso Aguirre

Clinical Trial Classification Problem (CTCP) is one of the cutting-edge real life applications in medicine.
The system described in this thesis aims to induce a classification model by *Clinical Trials (CT)* XML structured data in order to find a model that predicts patient's profile for eligibility criteria in breast cancer CT. Remarkably, the task considered in this work has been oriented to prediction of both *Eastern Oncology Group (ECOG)* and *Karnofsky (KPS)* scales. This scales represent the stage of the cancer disease patient suffers in order to be eligible to participate in the trial.
This particular task of CS comprise the use of *Natural Language Processing (NLP)* and *Machine Learning (ML)* techniques, which are two of the emerging areas of CS subfield's, *Artificial Intelligence (AI)*.

NLP has become one of the most important areas of AI due to its interesting cutting-edge applications in real life as *Textual Document Processing*, *Automatic Summarization*, *Machine Translation*, *Sentiment Analysis*, and many others.
Besides the great success NLP has achieved in big trendy and commercial applications now days, it has also been used to solve general text understanding tasks as *Natural Language Understanding*, *Natural Language Generation*, *Information Retrieval* and *Text Mining*, having a relevant impact in different real life fields such as Politics, Medicine, Finances, Governmental Security, Commerce, and Psychology.

Furthermore, ML an important branch of AI, comprise algorithms such as *Neural Networks (NN)*, *Support Vector Machines (SVM)*, *Ensemble Methods (EM)* that are based on data statistical learning. These methodologies have become meaningful to solve real-life classification and regression tasks. According to literature, *Non-Linear* models are suitable for complex tasks as *Speech Recognition*, *Drug-Drug Interaction Prediction*, *Twitter Sentiment Analysis* and *Clinical Trial ECOG-Classification* and instance of CTCP. On the other hand, *Linear* models are robust enough at applications as *Stock Market Trend Prediction*.

Clinical Trial ECOG-Classification (CTEC) is considered a computational task related to the decision support Systems utility. The task has a considerable importance in the field of medicine, in particular on the oncology department at breast cancer researching.
All this seems to be more workload than what the daily shift of an oncologist will allow. For this reason, a good approximation of an accurate decision support system that will help to asses the annotation of new breast cancer treatments patient's profile.
Consequently, this application as complementary support decision tool will help physicians to do work in parallel saving time at reviewing new extensive CT files. This may only require cancer specialist's custom-periodical reviews of the decision support model in order to validate the accuracy and performance.

Moreover, CTEC presents a high degree of difficulty in terms of the NLP and ML subtasks required for this application. Related to the NLP computational aspect, difficulties rely under the localization, classification and disambiguation of medical *Name Entities* such as: disorders, diseases, drugs, body parts, medical signs, etc. Furthermore, extraction of medical text related to measures, dose, units, etc. is also challenging. All these issues in *Electronic Health Records (EHR)* and CT have their origin on highly ambiguous acronyms and abbreviations, and on highly ambiguity of medical jargon. Professionals and technicians writing at the oncology department

and pharmaceutical research centers cause these language difficulties.

After described briefly, the AI trends and the CTEC application relevance and difficulty, let us describe technical aspects related to the problem. The data comprises 8,594 CT XML files related to different breast cancer treatments from all countries around the world. This information was retrieved from *clinicaltrials.gov* considered as the biggest and most popular source of trials data freely available in the medical scope.

Breast cancer CT treatments contain a relevant attribute, *eligibility criteria* that considers the stage of cancer in patient's condition in terms of a PS scale scoring. Most of the CT have an explicit PS scoring range on different breast cancer PS scales (ECOG, KPS). This particular application considers extraction of discriminant features and extraction of the PS scale scores to compose predictors and response variables of the training set.

As it was previously mentioned, the CS application requires extracting PS scale scores to compose response variables in training set by finding numerical patterns at eligibility criteria textual field. Moreover, the application requires extracting discriminant features from breast cancer CT eligibility criteria and consequently after both extractions builds a training set. After that, some additional processes have to be performed to data as data projections, removing noise, removing non-statistically representative samples, predictions refining, etc. All this in order to find the most suitable representation of the training set.

Consequently, after shaping up the data, the next step considers statistical model building, particularly to classify or predict the most suitable ECOG-KPS patient's profile. This profile, as previously mentioned resembles the participant profile based on their physical daily activity that constrains the application of participants for a new breast cancer treatment under clinical testing. The accurate prediction of this profile will result in avoiding misclassification assignments to trials and potential drug-side effects to patients.

In brief, the motivation of this project is selecting most suitable NLP and ML methodologies to minimize generalization error prediction i.e. increase prediction robustness over new CT samples.
Therefore, a robust model main advantage relies on finding the appropriate patient's profile for every breast cancer treatment and consequently reduces potential drug-side effects or health complications.
Further expectations are focused on development of a profitable software framework as complementary support medical system that may be used as a software tool for medical institutions and experts in the area, boosting the loads of work among the new breast cancer treatments profiling.

Acknowledgements

I would like to acknowledge my thesis advisors for the continuous support on the thesis from the very beginning before registration to the completion, for all the patience and for their friendship, Moltes Gràcies Horacio & Lluis…

Contents

List of Figures

List of Tables

List of Abbreviations

CTCP	**C**linical **T**rial **C**lassification **P**roblem.
AI	**A**rtificial **I**ntelligence.
NLP	**N**atural **L**anguage **P**rocessing.
ML	**M**achine **L**earning.
CT	**C**linical **T**rial.
FDA	**F**ood and **D**rug **A**dministration.
PS	**P**erformance **S**tatus.
ECOG	**E**astern **C**ooperative **O**ncology **G**roup scale.
WHO	**W**orld **H**ealth **O**rganization scale.
ZUBROD	**Z**ubrod scale.
KPS	**K**arnofsky scale.
LKS	**L**ansky scale.
EHR	**E**lectronic **H**ealth **R**ecord.
AD	**A**dmission **D**ate.
HPI	**H**istory of **P**resent **I**llness.
HC	**H**ospital **C**ourse.
DD	**D**ischarge **D**ate.

Dedicated to . . .

My parents Arnulfo & María Esther that have always supported me in my studies and work, specially to my mother that always believe in me in every aspect of my personal life..
Elisa & Benjamin that always show me what brothers mean..
Socorro & Josefina for loving me as mothers do. And the rest of my family for being there for me all the time..
Dario Principi, Jorge Rodriguez, Carles Coll, Zineng Xu, Xue He, Alberto Olivares, Guillermo Bernardez, & Bartosz Baran for being the best friends I could ever ask in my time at Barcelona..

Chapter 1

Introduction

1.1 Context

Clinical Trial Classification Problem (CTCP) is one of the cutting-edge real life applications in Medicine by the use of *Computer Science (CS)* methodologies.
The application presented in this thesis aims to induce a classification model that discriminates patient's profile in *Breast Cancer Clinical Trials (CT)*. The task considered in this work has been implemented as a multivariate regression predicting CT ECOG and KPS min/max scores.
This particular task of CS comprises the use of *Natural Language Processing (NLP)* and *Machine Learning (ML)* techniques, which are two of the emerging areas of CS subfield's, Artificial Intelligence (AI).

Medical informatics has gained relevance with the pass of the time through real life applications. All these applications are categorized by different utilities in medicine such as: clinical decision support, medical and pharmaceutical time line analysis, underlying patterns in medical entities relations, and many others.
For instance, clinical decision support system implementations [30] aim to aid decision making of health care providers and the public by providing easily accessible health-related information at the point and time it is needed. Moreover, medical and pharmaceutical reports time line analysis [10] purse to identify the temporal relations between clinical events and temporal expressions as *Admission Date*, *History of Present Illness*, *Hospital Course*, and *Discharge Date* in clinical reports. Additionally, Underlying patterns in medical entities relations [8] intend to find relationships among medical entities previously annotated and consequently asses relevance scoring among them. This application scope comprise different stages such as: *Semantic Tagging of Medical Categories*, *Relation Extraction in the Medical Domain*, *Ontology Metrics in the Medical Domain*.

Clinical Trial ECOG-Classification (CTEC) is considered a computational task related to the decision support systems utility type. Besides, the task has a considerable importance in the field of medicine, in particular on the oncology department at breast cancer researching. For this reason, the application has an important value as a decision support system for physicians in charge of breast cancer condition. The utility of the software application considers the following scenario at physician's daily loads:

- Oncologist load of work at daily schedule.
- Emergency department institutional prompts.
- Oncology department patient's follow up.

- CT pharmaceutical researching collaborations and design.

All this seems to be more workload than what the daily shift of an oncologist will allow. For this reason, a good approximation of an accurate decision support system that will help to asses the annotation of new breast cancer treatments patient's profile. Prediction of the scores related to patient's profile could be oriented in either ECOG or KPS scales. Consequently, this complementary support decision tool will help physicians to do work in parallel saving time at reviewing new extensive CT files. This may only require cancer specialist's custom-periodical reviews of the decision support model in order to validate the accuracy and performance. Despite the availability of a huge amount of CT reports, as shown in Table 2.1, Table 2.2, Table 2.3, and many of the documents lack ECOG or KPS scale information. So, it is important to enrich the document with an estimation of these scores.

Moreover, CTEC has certain degree of difficulty in terms of the NLP and ML tasks required for this application. Related to the NLP computational aspect, difficulties rely on the localization, classification and disambiguation of medical *Name Entities (NE)* such as: disorders, diseases, drugs, body parts, medical signs, etc. Furthermore, NLP aspects have certain percentage of difficulty at extracting medical text related to measures, doses, units, etc. All these NLP medical text issues in *Electronic Health Records (EHR)* and *Clinical Trials (CT)* have their origin on highly ambiguous acronyms and abbreviations, and on high ambiguity of medical jargon. Interestingly, different professionals and technicians writing at the oncology department and pharmaceutical research centers cause these language difficulties. Furthermore, issues on the ML subtasks are related to the learning itself of the model that will represent the CT - ECOG/KPS profiles accurately considering the most representative clinical text features or a combination among them together with an encoded representation of variable ranges associated to each CT. Therefore, finding a model that is able to predict or classify an ECOG or KPS profile, has to find a proper representation of the data. Moreover, a model itself has it's own intrinsic way of representing data (e.g. linear or non-linear), a complexity in terms of computational resources and own ways to penalize a high variance over the training CT samples. For this reason finding a proper representation of the data and a good approximation of an optimum configuration of model is not an easy task.

In essence, the motivation of this project is selecting most suitable NLP and ML methodologies and tools to minimize generalization error prediction i.e. increase prediction robustness over new CT samples. Therefore, a robust model main advantage relies on finding the appropriate patient's profile for every breast cancer treatment and consequently reduces potential drug-side effects or health complications. Further expectations are focused on development of a profitable software framework as complementary support medical system that may be used as a software tool for medical institutions and experts in the area, boosting the loads of work among the new breast cancer treatments profiling's.

In recent years NLP has become one of the most relevant areas of AI by it's interesting cutting-edge applications in real life as *Textual Document Processing*, *Automatic Summarization*, *Machine Translation*, *Sentiment Analysis*, and many others. Besides the great success NLP has achieved in big trendy and commercial applications now

days, it has also been used to solve general text understanding tasks as *Natural Language Understanding*, *Natural Language Generation*, *Information Retrieval* and *Text Mining*. Hence, general tasks applications in NLP require low-level NLP subtasks:

- *Localization and Extraction of Textual Content*.
- *Text Cleansing*.
- *Language Identification*.
- *Sentence Splitting*.
- *Tokenization*.
- *Stemming*.
- *Lemmatization*.
- *Morphological Analysis, including the normalization of dates, formulas, quantities, and units*.
- *Named Entity Recognition (NER)*.
- *Part Of Speech (POS) tagging*.
- *Lexical Analysis*.
- *Syntactic Analysis*.
- *Semantic Analysis*.

The application of these NLP functionalities have been impacted in meaningful applications among different life fields such as *Medicine, Finances, Politics, Governmental Security, Commerce, and Psychology*.

Furthermore, another popular AI branch, ML, aims to mimic the behavior of life phenomena's by building models for certain life processes based on data (observations). This recently applied area comprises algorithms that learn to either discriminate or predict in different ways how life phenomena's will conduct from a given input. Every ML algorithm is featured by:

- A learning approach: *Supervised*, *Semi-Supervised*, *Unsupervised*, or *Reinforced*.
- A learning task: *Classification*, *Regression*, or *Ranking*.
- A learning complexity: *Linear* or *Non-Linear*.
- A learning outcome: *Mono*, *Binary* or *Multiple*.

The most popular ML algorithms in terms of the number of applications registered in literature are: *Artificial Neural Networks*, *Support Vector Machines*, *Conditional Random Fields*, *Maximum Entropy* (also named *Logistic Regression*), *Bayesian Networks*, and *Ensemble Methods*. These methodologies have become meaningful to solve many of the real-life classification and regression tasks from the literature. Moreover, some of them are suitable for particular type of problems. According to literature, *Non-Linear* models are suitable for complex tasks as *Speech Recognition*, *Drug-Drug Interaction Prediction*, *Twitter Sentiment Analysis* and *Clinical Trial ECOG-Classification*. On the

other hand, *Linear* models are robust enough at applications as *Stock Market Trend Prediction*.

After a brief description of the AI methodologies deployed in this work, let us describe briefly the theoretical aspects of the CTCP and a simple description of the medical terms related to this application.

According to [28], *Clinical Trial (CT)* is research study that aims to analyze the effectiveness and safeness of drugs, strategies and devices applied in human health treatments. Moreover, it is considered an instrument to establish comparisons among different medical procedures to find the most suitable option for a particular disease or people's physical condition. Consequently, these analysis outputs are meaningful observations used to perform automatic decision making as complementary tools in the medical sector. CT data are obtained as a result of a strict research and collection methodology process, which involves the following stages:

1. *Laboratory Design*.
2. *In-Silico Analysis*.
3. *In-Vitro Analysis*.
4. *In-Vivo Analysis: Animal Testing*.
5. *In-Vivo Analysis: Human Testing*.
6. *Outcome Interpretation*.

There are different type of CT for different diseases and different treatments, however in this work, we strictly focus on CT *Breast Cancer*. Based on [9], CT in breast cancer are the safest and less invasive way to understand the disease for prevention and survival purposes. These types of CT are regulated by *The US Food and Drug Administration (FDA)*, an organization that aims to asses and approve new treatments as routine care in patients. The breast cancer CT and trials in general follow a rigorous study protocol. Besides, in some cases drug treatment proposals are constrained to different types of cancer drugs and others to potential combinations of drugs and procedures. Furthermore, before breast cancer CT results are analyzed, *The Human Testing* is conducted through Phases from 1-4, in which the number of people, and the complexity of the treatment is incremental at each stage until FDA approves the study (in case of the USA) and a consequent follow up is performed.

According to [14], researchers have found that in order to be able to conduct trials in a consistent way across all the research centers, hospitals and institutions they require a standard metric that asses the daily physical activities of a patient. This standard metric is known by researchers as *Performance Status (PS)* represented as a numeric scale, which tracks the profile of patients participating and asses their evolution through the experiment. However, performance status has been evaluated through the following different scales:

- *Eastern Oncology Cooperative Group (ECOG)*.
- *Karnofsky (KPS)*.
- *Lansky (LKS)*.

According to [41] ECOG scale was published by the *American Oncologist Charles Gordon Zubrod* and *The Eastern Cooperative Oncology Group* in 1982. This scale is also known as WHO or Zubrod and it comprises a range between 0 and 5, denoting 0 as a fully active pre-disease stage and 5 as death. Besides, this scale circulates and it is available at the public domain, and moreover, considered as future reference and further standardization criteria to assess functionality status of a patient. The main reason of this relies on the simplicity of the scale compared to other performance status scales.
Furthermore, another relevant scale is KPS [22], a PS scale that was published at first in literature by the American Oncologist *David Aryah Karnofsky* in 1948. The scale index ranges between 0 and 100, where 0 denote death and 100 normal activity minor signs of disease. The KPS scale main concern was the evaluation of patient's survival ability to chemotherapy.
Additionally, there is a third scale from the most well known scales, LKS performance status, which is consistently similar to KPS. The main difference relies in LKS main usage, as children quality observational scoring approximate system and assessing tool when an impediment of children to express their life quality exists.

The following table represents the equivalence among the different performance status scales scores, and a general description of each score. However, since the scoring and description of Karnofsky and Lansky scales are similar, we only consider comparisons among ECOG and KPS scales.

ECOG scoring	**KPS scoring**	**Patients Profile Description**
0	100	Normal, no evidence of disease.
	90	Normal activity, minor symptoms.
1	80	Effort required, some symptoms.
	70	Unable to perform active work.
2	60	Occasional assistance required.
	50	Considerable assistance required.
3	40	Disable, special care required.
	30	Severely disable, hospitalization indicated.
4	20	Very ill, hospitalization required.
	10	Moribund.
5	0	Dead.

TABLE 1.1: ECOG-KPS equivalences and patients profile description.

Consequently, after a wrap up of a detailed description related to the task's background considered in this work, the application relevance, and intrinsic and potential difficulties of the task, let us describe the basic technical aspects of the software implementation. The instance of the CTCP considered in this work was based from a *Stack Overflow Careers* job posting. Indeed, this document was distributed freely

among the candidates that applied for the data scientist position at *MedBravo* medical research center based in *Barcelona*. The implementation of the task [26] consists in applying NLP techniques over breast cancer CT Dataset retrieved from [11]. The task comprises a parsing ECOG scoring and a extraction of meaningful features from the clinical text of each file in order to build a training set. Therefore, after having a clean and representative dataset, the next step consists in deploying ML methodologies to build a good approximation of an optimum statistical model. After that, there is an implementation stage to classify the most suitable ECOG scoring ranges among testing set samples, where each sample is represented by a numerical vector, related to the relevance most representative terms from set of CT XML files. Besides, the sample class, which represents a range of classes, is represented as min and max scores from the range of classes assuming a consecutive multi class range.

1.2 Contributions

- Software Development[1]. *ECOG range-scoring XML CT extraction*. This module implements complex regular expressions and ECOG-KPS equivalence conversions.
- Software Development. *Cleansing, Tokenization, Stemming, Bag Of Words, and Feature Extraction over CT*. This module perform over all CT CORPUS, cleansing and extracting relevant features known as Bag Of Words from XML CT files.
- Software Development. *Machine Learning Regression-Classification Framework*. This module implements the most robust multi-regression and multi-classification, linear and non-linear models.
- Software Development. *Score Distribution, Prediction, and Aggregation Plot Framework*.
- Proposal of a suitable *Text Vectorization, Data Projections, Prediction Refining, Removal of Problematic Cases*, and *Model Section* for a *ECOG/KPS CT Breast Cancer Patient's Profile Prediction*.

1.3 Guideline

The document organization starts with *Chapter 1*, a brief introduction to a first insight of the work done in this thesis, this section additionally contain aspects as medical context foundations, project aims and goals, and contributions done to the medical application. After that, *Chapter 2* describes the computational and technical aspects of the task in terms of NLP and ML notions as a problem modeling. Additionally, we follow up with *Chapter 3*, a state of the art, in which we intend to grasp a wide panorama of the previous work done related closely to the application task of this project. Moreover, we continue with *Chapter 4* describing the experiments done with data representations, predictions refining, and model tuning in order to boost learning in predictions. After that, we present in closing results in a brief and a proper way in *Chapter 5*. To sum up, in *Chapter 6*, we established a conclusion of the experimental results obtained by the experimentation done in *Chapter 4*, we summarized all the work done, work trade-offs, and potential improvements as future work.

[1]Most of the developments have been done in Python, using specific modules for NLP and ML.

Chapter 2

Clinical Trial Classification Problem

2.1 Definition

As it has been mentioned in the previous chapter, breast cancer CT is a meaningful instrument that aims to analyze both the effectiveness and safeness of health treatments. This instrument also assesses strategies and devices related to patient's health care presenting breast cancer condition.

Additionally, as it was previously stated, PS is considered as the standardized metric to asses CT, track patient's treatment evolution and unifies researching analysis through different institutions and countries.
However, PS can be represented by different scales: ECOG, KPS, and LKS. These scales describe the stage of cancer based on the daily physical activity of the patients.

Therefore, we are going to consider the ECOG scoring from the scales mentioned, since it is the most standardized scale world wide for Non Neuro-oncological diseases. Furthermore, even if KPS scale is not the most standard metric it provides a wider range compared to ECOG and consequently more precision. For this reason we are going to consider the usage of KPS in experiments as a potential prediction advantage.
Moreover, related to the type of problem originally considered in this work, which was originally purposed as a multi-output and multi-label classification task type. After some approach discussions, we agreed to follow the respective wrap up adaptations:

- Consecutive numeric stages in PS scales.
- Prediction model seen multivariate regression, predicting *min* and *max* PS scale bounds in the range.

According to the generic NLP subtasks list included in *Chapter 1: Section 1*, task considered in this work comprise the following stages:

1. XML Data Acquisition.
2. ECOG scoring extraction (obviously in the case the score occurs explicitly in the text).
3. Data Cleansing.

4. Bag Of Words Construction.
5. Eligibility Criteria Text Vectorization.
6. SVD Data Projections.
7. Machine Learning Multivariate Regression.
8. Machine Learning Algorithms Tuning.

Before getting into task technical details, we proceed with proper reference to the data retrieved from *clinicaltrials.gov*. This data source is considered by medical professionals the biggest and most popular source of trials data freely available.
In order to reference this source, we are going to describe the data distributional aspects of the research scope retrieved by *clinicaltrials.gov* updated by March 2018:

- Distribution of the CT research approached by U.S. and other countries, shown in Table 2.1.
- Distribution of CT researching on breast cancer is performed by the U.S. and other countries, shown in Table 2.2.
- Distribution of the breast cancer study types considering CT from all the countries all over the world, shown in Table 2.3.

Country	**Registered**	**Recruiting**
U.S. only	95,265	17,702
Non-U.S. only	127,689	26,671
U.S. & Non-U.S.	14,615	2,390
Not provided	32,031	57
Total	269,600	46,820

TABLE 2.1: Registered and Recruiting General Clinical Trial Statistics.

Country	**Cancer**	**Breast Cancer**
U.S. only	31,268	4,316
Non-U.S. only	29,544	3,791
Total	60,812	8,107

TABLE 2.2: Cancer and Breast Cancer Clinical Trial Statistics.

To begin description with, the dataset [11] comprises 8,594 CT XML files related to different breast cancer treatments. For clarification purposes, let us mention that the 8,594 CT comprise the union of the unique 8,107 + 487 samples of the current data obtained by late March, 2018 and CT studies obtained at late May, 2015 that were removed on-line recently. Furthermore, for the sake of clarification purposes, there are 4,023 CT samples that contain PS KPS/ECOG scores in eligibility criteria CT XML field.

Study Type	Registered Studies	Studies w/Results
Interventional	6,573	866
Observational	1,515	46
Expanded Access	15	N/A
Total	8,107	912

TABLE 2.3: Breast Cancer (All Countries) Studies Types: Interventional, Observational, and Expanded Access in Clinical Trial Statistics.

Breast cancer CT treatments contain a *eligibility criteria* attribute, which considers the stage of cancer in patient's condition in terms of a PS scale scoring. *Eligibility criteria* is described in textual form but frequently includes explicitly the PS scoring. More details about the content of *Eligibility criteria* can be found in Section

2.2 Clinical Trial Fields

. This score ranges on different breast cancer PS scales (ECOG, KPS, LKS), and synonyms of them. However, the multivariate regression problem will focus on ECOG scale since most of the researchers consider it as an standard metric for Non Neuro-oncological type of cancer as breast cancer. Even though, KPS scoring is considered in the first stage of scoring extraction due its precise scale, so after predictions are obtained, there will be an equivalence stage to retrieve ECOG scores. Examples of an XML file with an *explicit* and *non-explicit* scoring can be found in *Appendix A*.

Furthermore, for this particular application we considered eligibility criteria field as the most relevant source for feature extraction based on the original requirements from [26]. Therefore, in this research we have used *eligibility criteria* as the clinical text source to build BOW.
Besides *Eligibility Criteria*, there other fields in the XML CT as *Brief Description*, *Title*, and many others that may have key discriminant information that can be extracted as potential features for boosting model learning.

2.3 Clinical Trial Scoring Distribution

The instance of the breast cancer CTCP considered in this work has a higher modeling complexity due to it's type of task and outcome. This particular instance of CTCP is featured by *Multi Label* and *Multi Output* in a classification learning type. This fact assumes that more than one class (PS cancer stage) can be present on CT participant's profile eligibility.
For this reason, we proceed modeling the task as a *Multivariate Regression* simplifying model complexity by its numerical label nature, considering two scores to learn *min* and *max* representing the CT PS range bounds.

In order to overview the scoring frequency distribution we performed one of the NLP generic subtasks *Regular Expressions (RE)*. RE were used to extract numerical scoring of PS scales from CT with an explicit scoring PS, keeping only the *min* and *max* values from the range.

Plots in Figures 2.1, 2.2, and 2.3 describe the frequency distribution of the range *min-score, max-score, and range-size* of all CT samples in CORPUS:

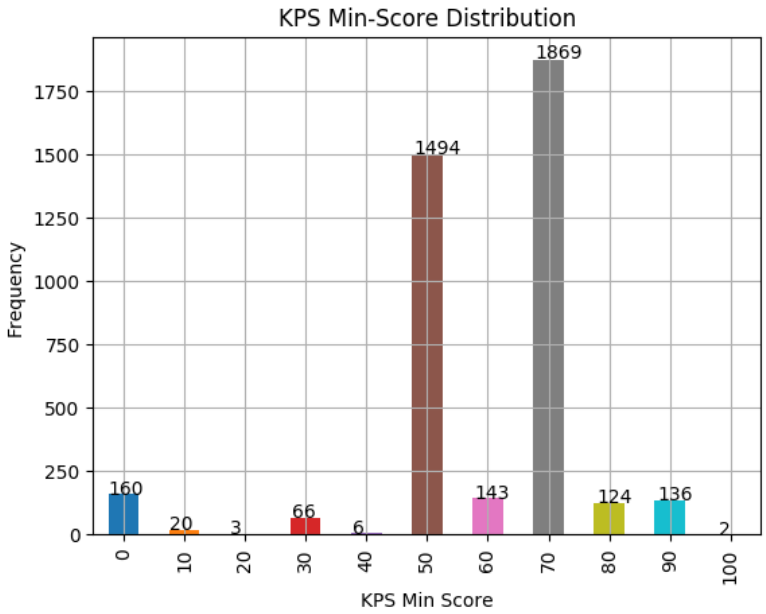

FIGURE 2.1: Min scoring distribution from KPS samples range.

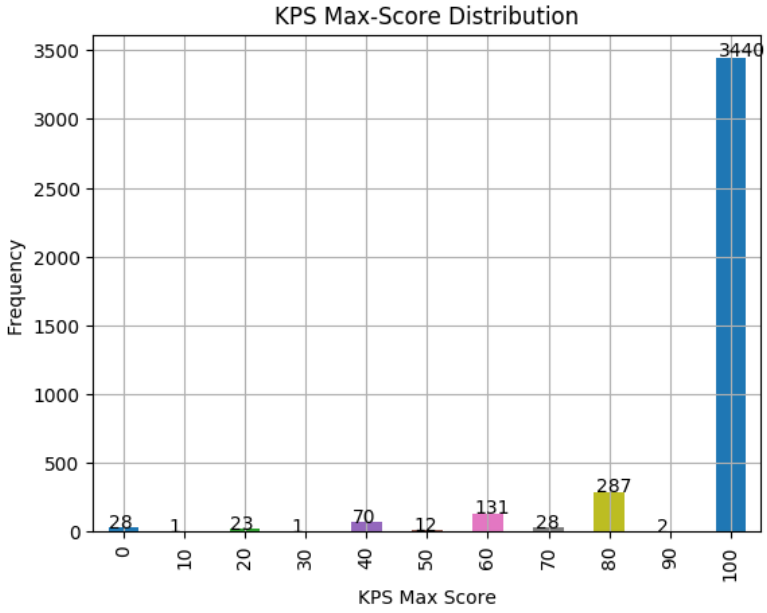

FIGURE 2.2: Max scoring distribution from KPS samples range.

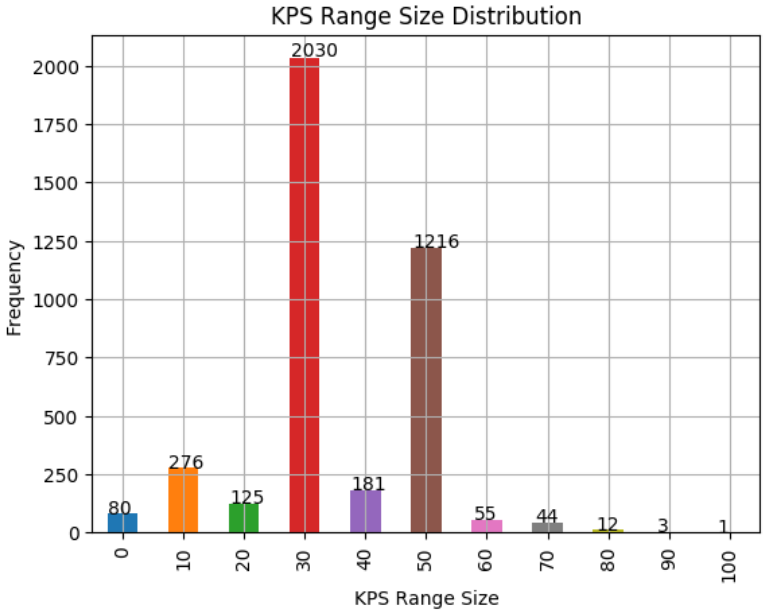

FIGURE 2.3: Range size (Max-Min) distribution from KPS samples.

As we can observe, breast cancer CT data for this particular annotation task seem to be class-unbalanced. We based this assumption on the data scorings reflected on the plots, this fact might be reflected as a higher complexity at model learning.

2.4 Clinical Trial Fields

CT reports are indented for human use, and, so, most of their content is textual. For this reason most of the features used for learning the models consists of words occurring in the document, regardless the order of occurrence, and the specific field where they occur. This model for representing the document is usually named a Bag of Words (BOW)[1]. Moreover, CT XML files are composed of different attributes or fields with information related to data source, sponsors, study summary, study outcomes, observations, eligibility criteria, study affiliations, location, and many others. However, the number of attributes contained in the CT vary based on the study and can be up to 60 different fields (leaves in the XML tree structure).

However, for this particular application we selected *Eligibility Criteria* field as *BOW* terms source by its direct relation describing the type of patient's profile that can be eligible as candidate in a numerical or textual way. Besides we considered this field for being consistent with the original task proposed by [26].

Furthermore, as future work we will be open to consider other relevant fields, which provide discriminant information for patient's profiling. For instance, CT XML fields as locations, affiliations do not provide any discriminant information to compose a BOW and consequently build a representative data set. Ideally, after performing a visual inspection of the attributes of XML files, we found that following XML fields may be considered as potential candidates of BOW source in further stages of the

[1]Other related forms of representation are Bag of lemmas, Bag of words+POS, bag of stems, and bag of n-grams

project. For sake of clarification, technical definitions are referenced by the *ClinicalTrials.gov - Protocol Registration Data Element Definitions for Interventional and Observational Studies* [31].

- **eligibility_criteria**: a plain text limited to 15,000 characters, which is denoted by a selection criteria list to filter participants in the clinical study. This list provided information in terms of inclusion and exclusion criteria and suitable for assisting potential participants in identifying clinical studies of interest. Remarkably, this items list includes both textual and non-textual information (as the age, and ECOG or KPS scores).
- **brief_summary**: plain text limited to 5,000 characters that describe a short title of the clinical study written for public distribution and understanding. This attribute should include information of the participants, conditions being evaluated, and interventions studied.
- **detailed_description**: a plain text limited to 32,000 characters, in which there is an extended description of the protocol, including more technical aspects compared to *Brief Summary* field. However, this field do not include the entire protocol and neither duplicate information recorded in other data elements as *Eligibility Criteria*.
- **brief_title**: plain text limited to 300 characters describing a short title of the clinical study in a public divulgation jargon. This attribute should include patient's information, condition being evaluated, and interventions studied.
- **study_type**: a text category, in which is described the nature of the investigation and usage for which clinical study information is being submitted. This attribute can be found by the following categories: interventional, observational and expanded access.
- **intervention_type**: a text category, describing the type of intervention studied in the clinical trial. The intervention types may be found as the following categories:
  - Drug including placebo.
  - Device including sham.
  - Biological-Vaccine.
  - Procedure-Surgery.
  - Radiation.
  - Behavioral as psychotherapy and lifestyle counseling.
  - Genetic including gene transfer, stem cell and recombinant DNA.
  - Dietary Supplement as vitamins, minerals.
  - Combination Product combining different intervention types.
  - Diagnostic Test as imaging, in-vitro and others.
- **minimum_age**: a numerical value, which indicates the minimum age a potential participant must meet to be eligible for the clinical study.
- **maximum_age**: a numerical value, which indicates the maximum age a potential participant must meet to be eligible for the clinical study.

- **mesh_term**: a text term belonging to the NLM's Medical Subject Headings[2] (MeSH), considered as a clinical term of a systematized medical nomenclature within the Unified Medical Language System[3] (UMLS) Metathesaurus.
- **condition_browse**: a text section, that contains a list of mesh terms.
- **condition**: a text term, which represents the name of the disease studied in the clinical trial.
- **phase**: a text category, which represents the numerical phase of the clinical trial related to the stages of a trial from the preliminary experiments to the analytic review of the advantages and disadvantages of a trial already approved by FDA. The following phases of the CT are:
  - **N/A**: not phases considered for behavioral studies.
  - **Phase 0**: a limited human exposure to clinical procedures.
  - **Phase 1**: a metabolism analysis after patient's drug reaction.
  - **Phase 2**: an evaluation of effectiveness of a drug.
  - **Phase 3**: a specific analysis of effectiveness after having preliminary evidence of effectiveness.
  - **Phase 4**: a study over FDA approved drugs to clarify risks, benefits and optimal use.
- **keyword**: a text term, which represent words or phrases that best describe the protocol. Keywords consider *The NLM's Medical Subject Heading (MeSH)* controlled vocabulary to keep a consistency in research writings. Additionally, keywords writing consider avoiding acronyms and abbreviations.

Furthermore, after flagged the XML field (*Eligibility Criteria*) from CORPUS as relevant to compose BOW, we proceed to perform the following NLP subtasks:

- *Eligibility Criteria* textual information extraction.
- *Data Cleansing* removing XML tags, filtering senseless symbols and term lowercasing.
- Text *Tokenization* (keeping terms with at least 2 characters) including *Lemmatization*.

Remarking the relevance of the NLP subtasks, we can denote that the main reason of using *Lemmatization* instead of *Stemming* is related with both techniques properties. Stemming algorithms work by cutting off the beginning or end of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. On the other hand, lemmatization takes into consideration the morphological analysis of the words, requiring language dictionaries queries. So *Stemming* is less precise, faster to compute but less error prone than *Lemmatization*.

Interestingly, lemma obtained by lemmatization, is the base form of all its inflectional

[2] https://www.ncbi.nlm.nih.gov/mesh
[3] https://www.nlm.nih.gov/research/umls/

forms, whereas a stem is not. Therefore, the reduction of the inflected words to the base form is performed more accurately by the morphological analysis of lemmatization rather than stemming.
Additionally, lemmatization seem to be suitable for this type of application since does not require keeping language verb tenses as other NLP application. Besides, another advantage of lemmatizing terms relies at reducing considerably the number of features in BOW, which have an important impact to data dimensions.

Given our BOW model for representing the document content, this limited linguistic process is enough for our needs, without using more sophisticated tools, as POS taggers, syntactic parsers, etc. After this step we are ready to collect all the stems for building the BOW representation.

2.5 Text Vectorization in Clinical Trials

After generating BOW from *CORPUS* and having samples truth values from the scoring extraction stage, we proceed to extract numeric features to compose the learning dataset. According to [33], the most representative *Text Vectorizers* are:

- *Count Vectorizer*.
- *Tfidf Vectorizer*.

Based on [33], *Count* and *Tfidf* vectorizers[4] represent a document as a numerical vector of n dimensions where each dimension corresponds to a word of the vocabulary BOW (a lemma in our case). So, n used to be large and the vector very sparse (i.e. having a big amount of zero values).

The difference between the two approaches is reduced to the way of weighting the components of the vector. After, establishing comparisons among both approaches, *Tfidf* weighting seem to represent term relevance in a better way than *Count*, which represent a raw frequency of a term in a document. Therefore, *Tfidf* approach has been considered represent CORPUS as set of vectors.

In short we are going to describe the *Tfidf* feature extraction approach [33], where the first denote the un-normalized calculation and the second the *L2 normalization* of the first one:

$$tf\_idf(t,d) = tf(t,d) * (1 + log\frac{1+n_d}{1+df(d,t)})$$

$$tf\_idf(t,d)\_norm = \frac{tf\_idf(t,d)}{\sqrt{\sum_{i=1}^{n_t} tf\_idf(t_i,d)^2}}$$

where:

d : A given document.

[4] Frequently count and tfidf vectors are normalized for getting a true probabilistic distribution, i.e. to transform the vectors into unitary vectors. Normalization simply consists on dividing each component by the summation of all of them

t : A given term in a d document.

n_d : Number of documents in CORPUS.

n_t : Number of terms in the document d.

t_i : i-th term in document d.

$tf(t,d)$: Raw occurrences of term t in document d.

$df(d,t)$: Number of documents where term t occurs.

Following this approach we obtained a sparse continuous matrix (i.e. with a higher presence of zeros), *Document Term Matrix (DTM)*.

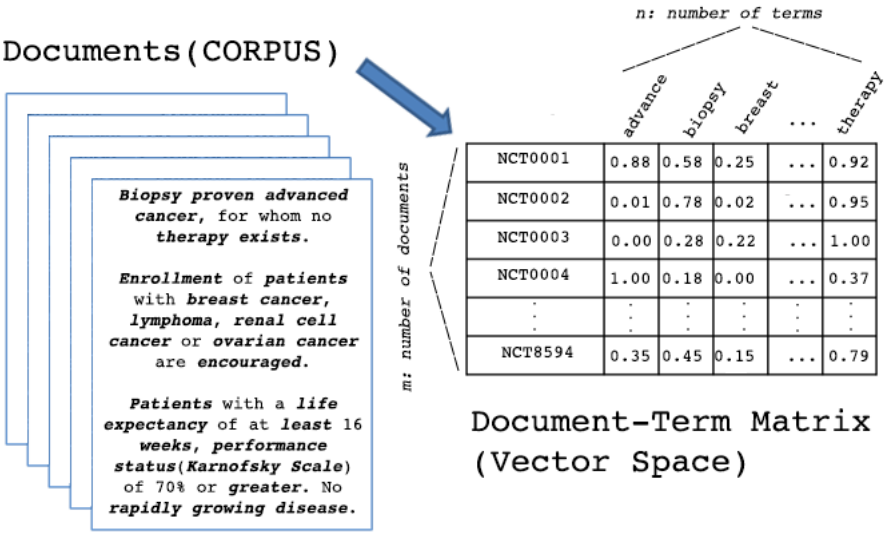

FIGURE 2.4: CT CORPUS representation as Document Term Matrix.

As we can observe in *Figure 2.4*, DTM columns represent the terms (lemmas) of BOW in alphabetic order, whether the rows resemble the documents in our CORPUS. Furthermore, in *Appendix B* can be found an example of the 25 most discriminant terms for every different configuration of *ngram_ranges*. This feature relevance evidence; consider combinations from 1 to 3 words in a single BOW term.

Getting into details, *Sklearn Vectorizers* [33] have interesting NLP properties:

- *ngram_range*.
- *min_df*.
- *max_df*.
- *max_features*.

The first parameter *ngram_range*, has to do with features representation, in which every BOW feature can contain one or more words in sequence (i.e. $mono-grams, bi-grams, tri-grams, .., n-grams$).

The second and third parameters *min_df* & *max_df*, stand for regulation of the

mathematical concept idf from the $tf - idf$ equation previously mentioned. The effect of adjusting these two parameters into lower-medium values consider only terms that appear with lower frequency in the CORPUS.

Consequently, after calculating the logarithm of a value higher than one, lead to influence proportionally positive the final $tf - idf$ score of the term considered at the calculation. Therefore, *Text Vectorizer* only keep the terms with a relevant idf value to build features data matrix.
Hence, in this application we consider the usage of different configurations, which are mentioned in *Chapter 5: Experiments*.

Finally, the last parameter $max\_features$, help to keep only the max number of features with the most higher $tf - idf$ value. Default parametric value, retrieves all the terms that can be possibly extracted. Interestingly, this parameter may be suitable in cases where the number of samples is lower and therefore, number of features must be constrained to a relatively lower value.
Besides, this parameters may be suitable, in cases when there is $n > 1$ at $ngram\_range$ values since the number of possible features increase highly.

After obtained a data matrix from the *Feature Extraction* stage, we obtained a higher dimension sparse matrix. The sparsity comprise the low or null relevance of specific terms for a particular CT based on the $tf - idf$ metric.

As it was previously commented, the number of lemmas in BOW represents DTM dimension. Particularly in our application after performing all the NLP subtasks previously mentioned, we constrained the maximum number of lemmas as *15,296*, for every $ngram_combination$, which is the maximum number of mono-gram features found by the vectorizer.
As we can observe, the number of features exceed the number samples available, for this reason we decided to perform *Data Projections*, i.e. *Dimensionality Reduction* techniques. In short we are going to describe how data projections can be done performing features mapping.

2.6 Data Projections (Features Mapping)

This feature selection approach aims to find a useful representation of the data, mapping features to a more representative numerical space. This approach does not involve a manual removing of the features but a projection of them that most of the time implies a linear and non-linear combination among them. These type of methods help to reduce data dimensions, by finding the k significant features as a projection of the originals d. The main advantages of dimensionality reduction are:

- Finding a mathematical representation within which you can describe most but not all of the variance within your data, retaining relevant information but reduced to a considerable size to ease model induction.
- Preserving its Euclidean structure but does not suffer from curse of dimensionality (i.e. data become highly sparse making unfeasible learning for ML algorithms that require statistical significance).

This analysis is usually performed by the general real matrix factorization method, *Single Value Decomposition (SVD)* [36] [5]. SVD is considered a generic method that

can obtain PCA after performing matrix factorization of a covariance matrix.

There are many implementations available of LSA, PCA and related approaches. From these I have chosen SVD-Truncated, included within Sklearn for the following reasons:

- *SVD-Truncated* robustness compared to PCA, since it can be applied to distance and similarity matrices.
- *SVD-Truncated* robustness to deal with sparse data from a non-numerical nature, (e.g. lemmas encoded as numerical values) in comparison with PCA.
- *SVD-Truncated* combines features (lemmas) values having close meaning by means of weights resulting as robust to *synonymy* and *polysemy* by grouping similar features.
- *SVD-Truncated* cheaper numerical calculations over sparse data, since it does not standardize data before computing SVD, this result as an efficient performance.
- *SVD-Truncated* framework is suitable for matrices formed with *Count* or *Tfidf* vectors.

The *SVD-Truncated sklearn* configuration parameters are: *n_components*, *algorithm*, *n_iter*, and *random_state*. The first parameter stands for the desired dimensionality of the projection. The second parameter consider the algorithm under the projection task, either *ARPACK* or *Randomized*. The last two parameters depend highly on the algorithm implemented. Remarkably, we have to point out that the components and the projection obtained are sensible to the algorithm and random state properties.

2.7 Clinical Trial Prediction Evaluation Metrics

After obtaining a representative dataset, we continue describing the error metrics considered by the *Machine Learning (ML)* models implemented in this task. For the sake of proper clarifications of the solving approach, as it was previously mentioned, we considered to accomplish the multi-output, multi-label classification task as multivariate regression task taking as an advantage a potential representation a numerical range of classes as two numerical values representing the *min* and *max* values of a range of numbers comprising ECOG or KPS scales.

For this reason, we considered regression predictions metrics in this task. According to literature the most well known regression error measures are: *Mean Squared Error (MSE)* and *Scaled MSE* ($1 - R^2$). Remarkably, $1 - R^2$ metric intends to explain how big is the prediction error in proportion with the variance of the truths. Based on [32], this metric is commonly used as model performance indicator, in which lower values mean higher performance.
Consequently, we are going to consider MSE and $1 - R^2$ metrics to asses error and proportion of response variance captured by the model.

To notice, the usage of this metrics consider a multivariate regression task, in which we intend to predict two values *min* and *max* scores from breast cancer CT patient's profile range.

$$MSE(X, y_j) = \frac{\sum_{i=1}^{n} (f(x_i) - y_{i,j})^2}{n}$$

where:

X : all features from all the samples in data.

y_j : j-th component from all the samples responses values (integer vector).

n : number of samples in data.

x_i : all features from i-th sample (continuous vector).

$y_{i,j}$: j-th response value component of i-th sample (integer value).

Range : range values depend on the y-domain values. Being 0 an ideal value.

$$VAR(y_j) = \frac{\sum_{i=1}^{n} (y_{i,j} - \bar{y_j})^2}{n}$$

where:

y_j : j-th component from all the samples responses values (integer vector).

n : number of samples in data.

$\bar{y_j}$: Mean value of the j-th component from all the samples responses values (continuous value).

$y_{i,j}$: j-th response value component of i-th sample (integer value).

$$1 - R^2(X, y_j) = \frac{MSE(X, y_j)}{VAR(y_j)}$$

$$1 - R^2(X, y_j) \in [0, 1)$$

where:

X : all features from all the samples in data.

y_j : j-th component from all the samples responses values (integer vector).

Range : range values normally oscillate between 0 and 1, however when the MSE is higher than VAR(y), values may be higher than 1. The lower the value the higher performance of the model.

2.8 Clinical Trial Multivariate Regression Complexity

In this section, we describe briefly the mathematic complexity behind CT multivariate prediction problem in term of matrices operations. Interestingly, understanding about task complexity always give descriptive insight of how costly is to perform

a ML task in terms of computational resources. This analysis, is often required to asses how difficult can be scaling the problem to big data scenarios.

For this analysis we are going to consider the following elements:

- Features matrix X: features data matrix extracted after performing *Text Vectorization*. Matrix dimensions are (n, m), in which n represent the number of samples and m the number of features of BOW retrieved from *Text Vectorization*.
- Response matrix Y: response data matrix. Matrix dimensions are $(n, 2)$, in which n represent the number of samples as X, and 2 the output variables min and max ECOG/KPS scores respectively.

For the sake of description, the experiments described in this research n is set to $4,023$ and m to $15,296$. These values represent the samples with explicit ECOG/KPS scoring, and the maximum number of unique single lemma terms in CORPUS.
For this analysis, we would consider a asymptotical complexity [25] [19] based on the regression normal equation and assuming that the number of samples is bigger than the number of features (e.g. $n > m$).

Task	**Operations**	**Time Complexity**
SVD	Overall	$O(min(mn^2, m^2n))$
Regression	X^tX	$O(m^2n)$
	$(X^tX)^{-1}$	$O(m^3)$
	X^tY	$O(m^2n)$
	$(X^tX)^{-1}X^tY$	$O(mn)$
	Overall	$O(m^2n)$
MSE	$X((X^tX)^{-1}X^tY)$	$O(m^2n)$
	$Y - (X((X^tX)^{-1}X^tY))$	$2n * \Theta(log(d))$
	Overall	$O(m^2n)$

TABLE 2.4: CT multivariate regression task time complexity.

2.9 Learning Algorithms

In this section, we describe the algorithms implemented in the multivariate regression task considered in this work. For this application we found interesting the fact of comparing ML predictors with a different complexity to represent the tendency of data (i.e. linear or non-linear) way to fit data. Considering the type of prediction

task and data properties as high-dimensional spaces after extracting features at *Text Vectorization*, we considered that the most appropriate candidates for the task given the previously mentioned conditions are: *Partial Least Square (PLS)* as a linear model and *Multi-Layer Perceptron (MLP)* neural network as a non-linear model.

2.9.1 Partial Least Square (PLS)

Proposed by the Swedish Statistician, Herman O. A. Wold in 1975 [17]. The algorithm is featured by the following specifications:

- A bi-linear factor model.
- A latent variable approach related to PCA.
- Finds a linear regression over X and Y projections into a new space.
- Suitable when there are more features than samples in the dataset.
- Suitable when there exist multi-collinearity among predictor variables X.
- Applied frequently at chemo metrics, bioinformatics, anthropology and neuroscience analyses.

The PLS algorithm is represented by the mathematical expressions, in which X and Y represent features and response matrices from a given dataset:

$$X = TP^T + E$$

where:

n : Number of samples in data.
m : Number of features in data (after dimensionality reduction).
X : Features data, (n, m) data matrix.
T : Numerical mapping of X, (n, l) data matrix.
P : Weights coefficients (m, l) matrix.
E : Error (n, m) matrix, assuming independently random distribution.

$$Y = UQ^T + F$$

where:

n : Number of samples in data.
p : Number of response variables in data.
Y : Response data, (n, p) data matrix.
U : Numerical mapping of Y, (n, l) data matrix.
Q : Weights coefficients (p, l) matrix.
F : Error (n, p) matrix, assuming independently random distribution.

X and Y decompositions aim to maximize the covariance between T and U in such a way that projected variables of X are correlated with projected variables of Y, showing consistency among features and responses on the new projected space.

2.9.2 Multilayer Perceptron (MLP)

As an alternative to the bi-linear model PLS we decided to try a non-linear model from the family of Neural Network (NN) model. Being the size of our dataset for learning rather small, it is out of consideration to use a deep learning model, that need a big dataset for learning. We decided, so, to move to a simple Feed Forward approach, specifically a Multilayer Perceptron (MLP). MLP has its foundations on former research related to *Single Layer Perceptron* by McCulloch and Pitts in 1940, the first model symbolized a neuron that worked as a linear model combining inputs with respective weights. This model was applied either as binary classification problems that required threshold and step functions or as a regression task that applied a simple linear combination considering bias. Based on the fact that a unique neuron was not capable to discriminate among non-separable data, and after some studies in 4 decades, in 1985 McClelland and Rumelhart proposed neuron based improved model that was capable to track non-linear problems, Multilayer Perceptron [34]. The MLP algorithm is featured by the following specifications:

- A non-linear model.
- An abstraction of cognitive and neural learning and connectionism among neurons.
- An architecture where all the neurons of the network are fully connected.
- A feed-forward approach, where learning starts from the first hidden layer and goes through the other hidden layers to the output layer, with no cycles or loops.
- Neurons have non-linear activation functions, (i.e. inputs come through a non-linear function).
- Several neuron activation functions could be used: *tanh*, *sigmoid* (logistic), *relu* (semi-linear) as shown in *Table 2.5*.
- Network weights are learned by back propagation approach, which refines weights by a gradient descent error respect to networks weights modulated by a learning rate. The most popular learning algorithm for this kind of network is the Stochastic Gradient Descent, .
- A typical network architecture comprises inputs, a hidden layer and an output layer. This fact is based on *The Universal Approximation Theorem*, in which a continuous function that maps intervals of real numbers can be approximated arbitrarily closely by *Multilayer Perceptron* with just one hidden layer.

The algorithm considers the following expressions related to delta rule considering the X(n,m), Y(n,p) dataset features and responses respectively and z neuron input:

Name	Mathematical Expression
Identity	$f(z) = z$
Rectified Linear Unit (Relu)	$R(z) = max(0, z)$
Sigmoid (Logistic)	$\sigma(z) = \frac{1}{1+e^{-z}}$
Hyperbolic Tangential	$tanh(z) = \frac{2}{1+e^{-2z}} - 1$

TABLE 2.5: MLP Activation functions.

MLP Delta Rule for Iterative Learning MLP algorithm comprises an optimization algorithm that seeks to minimize error, adjusting weights values to approximate response variables. The optimization method implemented is gradient descent, which calculates the weights with minimum error by obtaining the partial derivate of the weight errors respect to the weights.

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}(Y_{i,k} - f(X_i))^2 = \sum_{i=1}^{n}[Y_{i,k} - g(\sum_{j=1}^{d} w_j X_{i,j} + w_0)]^2 | \forall k \in p$$

where:

$E(w)$: weights error function.
i : index of i-th sample.
n : Number of samples.
j : index of j-th neuron.
d : Number of neurons.
k : index of k-th response variable.
p : Number of response variables.
$Y_{i,k}$: k-th response value of i-th sample.
$f(X_i)$: Hypothesis value of the i-th sample.
$g(X, w)$: g activation function.
w_j : j-th neuron weight.
$X_{i,j})$: j-th feature of the i-th sample.
w_0 : Threshold/bias.

In order to obtain minimum error derivative has to be calculated obtaining the following expression:

$$\frac{\partial E(w)}{\partial w_j} = -\sum_{i=1}^{n}(Y_{i,k} - f(X_i))g'(w^T X_i)X_{i,j} | \forall k \in p$$

where:

$\frac{\partial E(w)}{\partial w_j}$: partial derivate of weight errors respect to the j-th neuron weight.
i : index of i-th sample.

n : Number of samples.

j : index of j-th neuron.

k : index of k-th response variable.

p : Number of response variables.

$Y_{i,k}$: k-th response value of i-th sample.

$f(X_i)$: Hypothesis value of the i-th sample.

$g'(X, w)$: Derivate g activation function.

w_j : j-th neuron weight.

$X_{i,j})$: j-th feature of the i-th sample.

Consequently if g activation function is the identity from *Table 2.5* we obtained the Least Means Squares (LMS) function:

$$\Delta w_j(Y_k) = \alpha \sum_{i=1}^{n} (Y_{i,k} - f(X_i)) X_{i,j} | \forall k \in p$$

where:

$\Delta w_j(Y_k)$: increment of j-th weight considering k-th response variable.

α : learning rate.

i : index of i-th sample.

n : Number of samples.

k : index of k-th response variable.

p : Number of response variables.

$Y_{i,k}$: k-th response value of i-th sample.

$X_{i,j}$: j-th feature vale of the i-th sample.

$f(X_i)$: Hypothesis value of the i-th sample.

These techniques represent a linear Regressor where weights are estimated iteratively the most applied learning rule according to the literature. After this formulation, the same expression can be adapted for iterative algorithmic considering t number of iterations, with a variable learning rate as the following expression:

$$\Delta w_j(t) = \alpha_t (Y_{i(t)} - f(X_{i(t)})) X_{i(t),j}$$

where:

t : index of t-th iteration.

$\Delta w_j(t)$: increment of j-th weight at t-th iteration.

α_t : learning rate at t-th iteration.

$Y_{i(t)}$: i-th sample response value at t-th iteration.

$X_{i(t),j}$: j-th feature vale of the i-th sample at t-th iteration.

$f(X_{i(t)})$: hypothesis value of the i-th sample at t-th iteration.

2.10 Pros & Cons of Solving Methodology

As every formal scientific research, a balance between the advantages and disadvantages of the methodology proposed in this thesis is required in order to observe the strengths and particularities of the contribution presented in this work as the vulnerabilities or potential improvements in term of performance and accuracy at predictions of the *Clinical Trials (CT)* profiles considering ECOG/KPS scales.

Advantages of NLP & ML Methodology

- NLP robust framework to perform extraction of the response values of dataset represented as ranges bounds *min* & *max*. Note that, when present, this information occurs in a variety of formats.
- NLP robust framework to perform text cleaning and pre-processing *Stop word-Removal, Stemming, Tokenization* in order to preparing the text for further feature extraction.
- NLP robust framework to perform *Text Vectorization* to explore different representations of features as n_grams combinations, keeping the most representative features based on $tf - idf$.
- ML sparse robust *Single Value Decomposition* method to project features extracted by *Text Vectorization* to a more feasible separable space.
- ML suitable *PLS* bi-linear factor model to predict multivariate ECOG/KPS scores, in high dimensional data, in which the number of samples is lower than the number of features.
- ML suitable *MLP* non-linear model to predict multivariate ECOG/KPS scores, in high dimensional data, feasible to generate data distribution of non-separable data.

Disadvantages of current NLP & ML Methodology The proposed approach presents, however, some limitations that could be addressed as future work.

- NLP potential improvement considering specific medical NLP annotation tools as *Metamap*[7], *NCBO annotator*[3], *CTakes*[35], *Snow Med*[37], etc. instead of the general ones we have used. These tools may be useful to extract meaningful features as medical entities, drugs, and treatments that could represent data predictors in a more accurate way.
- NLP potential improvement considering *Embedding* of medical text. Embedding, comprise mapping the sparse input vectors into dense, low dimensional space. This adecuations can be performed at word level as Word2Vec[1], Glove[29], for general language. Furthermore, Pyysalo[4] is suitable for the medical domain and complex language contexts. In fact embedding combines vectorizing and dimensional reduction into one single step.
- ML potential fail in *PLS* assumption, in which PLS is profitable when there is multi-collinearity in X predictor variables (i.e. some of the features have a higher correlation among them).

- ML potential fail in *MLP* consideration, in which a lower number of samples compared to the number of features leads model to over fit training data failing to capture the statistical generation of the data. This may be solved by implementing *early stopping*, *drop out*, or other criteria on MLP.
- ML potential fail in *MLP* back propagation property in terms of the speed of convergence, the optimization algorithm may fall into local minima if speed of convergence is higher. This may be solved using a low or adaptable *learning rate* in MLP.
- NLP & ML potential improvement considering connection frameworks with *Hadoop, Map reduce, a MPI-Python* to parallelize the task if the problem is scaled to big data, though it is not the case in the problem at hand.

Chapter 3

State of the Art

Now a day, medical informatics has gained a higher importance through real life applications. All these applications are categorized by different functionalities in medicine such as: clinical decision support systems, medical and pharmaceutical time line software analysis, underlying patterns in medical entities relation's analytics, and many others. All these medical applications by computational resources have brought more resources to extend life expectancy, safeness and comfort on human beings.

However, these real life applications face their own challenges and difficulties as:

- Localizing, classifying and removing ambiguities on medical *NE* from a high variety of genres, some of them specially challenging as *Electronic Health Records (EHR)* and *Clinical Trials (CT)*.
- Extracting medical text related to measures, doses, units from these documents.
- Finding discriminant and compact representations of medical data.
- Tuning Machine Learning algorithms to such materials.
- Ensuring classification/prediction higher accuracy to give proper and accurate results.
- Solving computational performance issues.

Therefore, relevant and profitable medical application has a lot of multi disciplinary work behind them and consequently require dedicated work and research always with an innovative and modern vision of life needs.

According to the most recent surveys from medical informatics, the cutting-edge applications are: *Clinical Decision Support (CDS)* [13], *Medical Question Answering (MQA)* [16], *Finding Patterns in Annotation graphs* [6], *Semantic tagging of medical categories* [12], *Terminology Extraction for the Medical Domain* [38], *Metrics in Ontologies in the Medical Domain* [27], and *In silico analysis of drugs* [20].

Furthermore, related to the medical resources, the most prestigious and trustable freely available data banks of clinical trials are *ClinicalTrials.gov* [11] and *Linked CT* [21]. First of them store trials from all around the world and the second source is composed by trials of different sources including the former *ClinicalTrials.gov* and is built in generic and basic organization and visualization tools. Moreover, related to ontologies, there are relevant and well know references available from *BioPortal* [18].

Many BioMedical corpora are, too, publicly available [40]. Finally, another meaningful resources related to the medical informatics are biomedical processors as *NCBO annotator* [3] , *Metamap* [7], and *CTakes* [35] that ease the medical *NE* recognition task resulting as a potential alternative for categorical features generation.

Furthermore, limiting medical informatics research to clinical trials applications there are applications strongly related to the problem considered in this thesis.

One of these applications consider *Karnofsky Performance Status (KPS)* deploying an algorithmic system for its evaluation. In [30], *KPS* is considered important as *Clinical Trial (CT)* assessment criteria.

In this contribution there is an overview scale basis as a matter of discussion of its subjectivity in the 1940's, where cancer physicians and mental health professionals found a higher correlation between cancer evolution and psychosocial aspects in patients. Consequently after some researching, health professionals found 7 psychosocial factors that empirically explain cancer evolution applying a multivariate regression analysis. Notably, these factors now comprise part of the KPS scale-stage descriptions. Moreover, this research established comparisons among other scales such as ECOG and AKPS to KPS highlighting KPS advantages and no improvements by the others. In their solving methodology, they proposed a *Decision Tree* approach as a CT-KPS classification model.

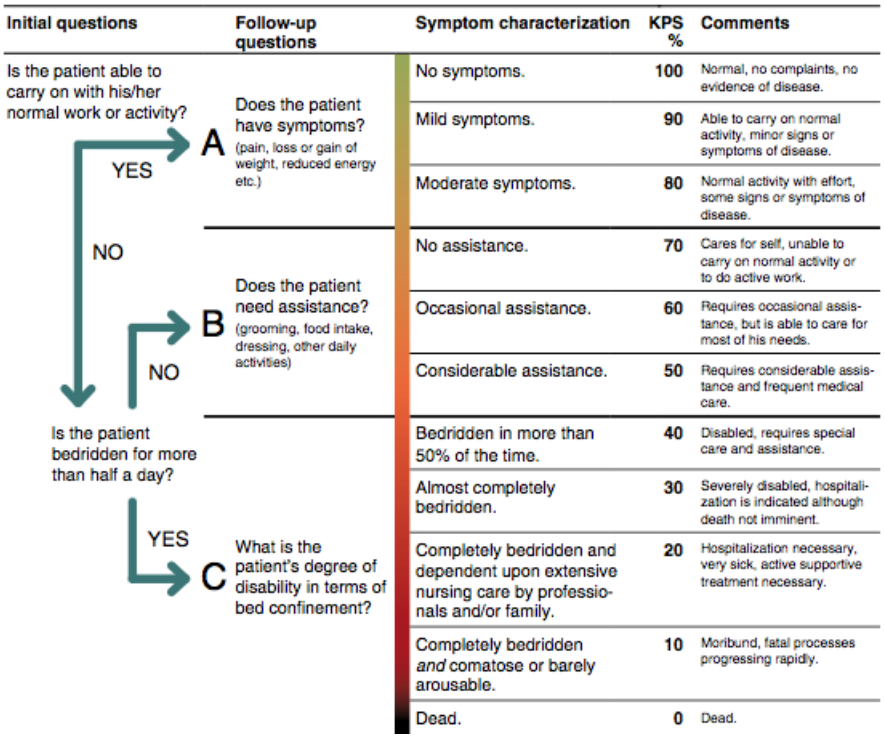

FIGURE 3.1: Decision Tree CT-KPS Classification Model from [30].

According to the authors the system had a good performance for this type of application, but they remark potential drawbacks and limitations faced by their CT-KPS model. In short we are going to describe briefly the limitations of the model as part of possible future improvements for similar applications:

- The model assumes fixed and term-constrained inputs queries (textual/oral) classification but has a lack of robustness against synonymy, polysemy and other language ambiguities.
- The model is not robust against fuzzy input queries, e.g. "normal, no complaints, no evidence of disease", where the third term (i.e. no evidence of disease) inputs fuzziness in the query, since patient may had previous cancer experiences or family incidence, and as consequence system do not consider potential fuzzy queries and outcomes.
- The model does not consider drug's name entities or drug's categories as data features in the classification model.
- The model is not robust at predicting KPS range of stages, therefore is constrained to one stage KPS classification, and consequently it has a non-realistic behavior according to CT KPS real profiles.
- The model is not induced by data inference, therefore they are not considering the most updated features from *ClinicalTrials.gov* CT XMLs.
- There is not a defined evaluation for this classification model.

Moreover, another relevant clinical trial application is focused on performing data mining over cancer vaccine trials. In this research [10], the authors try to enhance the motivation of why collaborating with cancer through particular statistics from *The World Health Organization's Global Burden of Disease (WHOGBD)*.

This work considers the gathering of relevant resources as bio-molecular ontologies from *The National Center for Bio-technology Information (NCBI)* and as cancer vaccines CT data from *The US National Cancer Institute (NCI)* through *ClinicalTrialas.gov*.

The work proposed a visualization framework that aims to understand the cancer vaccines CT medical sector by extracting, summarizing, and generating demographics, statistics and plots from specific fields in the CT XML files. The final usage of this work highlights relevant aspects as cancer vaccines timeline usage, survival percentages for different cancer types, and a distribution overview of treatments among institutions all these bring as consequence a clear clinical and pharmaceutical advantage in the market coverage.

Other applications of trials consider the implementation of medical annotators. An example of this application utility[8] is the use of *NCI Thesaurus* to annotate CT and graph representation of entities relations for data mining processes.

The annotations comprise clinical entities tags among CT conditions, drugs, etc. According to the authors, relevant relationships and patterns can be extracted from annotated CT. The proposed framework considers the following stages:

1. Gathering data from an open link source *LinkedCT.org* [21] *ClinicalTrial.gov* [11] XML CT data bank.
2. Implementation of ontologies as *NCI Thesaurus* & *NCI Methasaurus* to tag conditions, drugs, etc. with medical entities.

3. Feeding a fully connected graph with all the medical entities tagged in all the CT.

4. Performing *Dense Sub Graph (DSG)* generation to remove less relevant relations in terms of a taxonomic distance metric, in order to keep a compact but informative representation of the graph.

5. Performing *Graph Summarization (GS)*, representing graph as partitions of super-nodes and super-edges, after edges addition-removal that encode more efficiently the underlying relations of medical entities.

Another relevant application in CT is the automated extraction of clinical trial characteristics from medical scope text [23]. Considering CT as the most important sources of evidence for guiding evidence-based practice and the design of new trials, most of this information is available only in free text (e.g. in journal publications). This denotes the intensive labor to process text for systematic reviews, meta-analyses, and other evidence synthesis studies.

[23] propose a framework that assists users with locating and extracting key trial characteristics (e.g. eligibility criteria, sample size, drug dosage, primary outcomes) from full-text journal articles reporting on randomized controlled trials (RCT).

The framework proposed consider the following modules:

- An information extraction (IE) engine that searches the article for text fragments that best describe the trial characteristics.

- A web browser-based user interface that allows human reviewers to assess and modify the suggested selections.

The IE engine implements a statistical text classifier to locate those sentences that have the highest probability of describing a trial characteristic. After that, the second module applies simple rules to these sentences to extract text fragments containing the target answer. As a research extension, in *The Human Studies Database (HSDB) Project*, there are federating the computable description of trial design, execution, and results to support large-scale data analysis and synthesis across many ongoing and completed studies.

Additionally, according to [2] eligibility criteria extraction problem has been recognized as a relevant functionality for CT data mining. This work considers the fact that eligibility criteria fields in clinical trials are represented as free text, therefore, their automatic interpretation and the evaluation of patient eligibility is challenging.

The approach proposed in this work considers the identification of contextual patterns and semantic concepts that together define the machine-interpretable meaning. Consequently, their aim is to find the most relevant concepts occurring in eligibility criteria that need to be mapped to patient record to enable automatic evaluation of patient eligibility. The research considers exploring the concepts that occur in eligibility criteria related to a particular disease. And as a consequence identified concepts will be used to link to corresponding data items in patient record, to enable evaluation of patient eligibility.

Finally, there is another recent application [15] related to CT focused on evaluation of the use of document similarity methods to retrieve unreported links between *ClinicalTrials.gov* [11] and *PubMed* [24]. In this work, they considered extraction of terms and concepts of 72,469 structured XML files from *ClinicalTrials.gov* and 276,307 registry entries from PubMed. Besides, they implemented tested methods for ranking articles across 16,005 reported links and 90 manually-identified unreported links.

The research considered the usage of distance metrics as *Euclidean distance, Cosine,* and *Jaccard*. Moreover, performance was assessed by the median rank of matching articles, and the proportion of unreported links that could be found by screening ranked candidate articles in order.

Chapter 4

Experiments

In the following section we describe the experiments done in this work. These experiments comprise different tuning configurations from multiple parameters of the different steps of prediction task namely:

- Text Vectorization.
- Data Projection.
- Prediction Refining.
- Sample Problematic Cases Removing
- Model Tuning.

As stated in *Chapter 2*, we will compare two different regressors, a bi-linear *PLS* and a non linear *MLP* in order to find the most appropriate type of model for the particular medical considered in this work.

Furthermore, we found these two particular models suitable considering their property of predicting multi-output values. Considering the previously mentioned wrap around from classification to prediction in task we must consider both models *PLSRegressor* and *MLPRegressor* to solve the regression task.

However, we started the experimental results analysis considering default parameters in both algorithms that will be subject to tuning in further stages of the experimentation. Therefore, let us describe now the default parameters of both models based *Python Sklearn* framework.

1. *PLSRegression* parameters:
   - $n\_components = 2$
   - $copy = True$,
   - $max\_iter = 500$
   - $scale = True$
   - $tol = 1e-02$
2. *MLPRegressor* parameters:
   - $solver =' lbfgs'$
   - $alpha = 1e-05$
   - $activation =' relu'$
   - $hidden\_layer\_sizes = (10, 1)$
   - $tol = 0.0001$
   - $max\_iter = 200$

- $random\_state = 99$
- $early\_stopping = False$
- $warm\_start = False$

4.1 Dataset specifications

After a brief descriptions of the algorithms implemented in this work and a previous description of data in *Chapter 2*, we will reference data source to $clinicaltrials.gov$.The biggest and most popular source of trials freely available. Remarking that dataset was updated by the last CT available March 2018.

Country	**Cancer**	**Breast Cancer**
U.S. only	31,268	4,316
Non-U.S. only	29,544	3,791
Total	60,812	8,107

TABLE 4.1: Distribution of CT researching on cancer/breast cancer performed by the U.S. and Non-U.S. countries.

Additionally, we must mentioned that the testing/training slicing framework was based on *10-Fold-CV*, in which we train and predict 10 times using 90% and 10% for training and testing respectively.
Furthermore, let us describe the particularities of the data used to train models for a KPS and ECOG PS prediction types. Hence, as it was previously mentioned in *Chapter 2* the samples we only considered for training set (4023) have the ECOG/KPS explicit score ranges in eligibility criteria.

Performance Status	**Samples**	**Features**
KPS	3,767	15,296
ECOG	4,023	15,296

TABLE 4.2: Dataset sizes considered for KPS and ECOG predictions.

As we may observe the samples used on each PS scale prediction vary, considering that KPS predictions generalize better without problematic samples, which are not statistically relevant. Although, ECOG predictions are induced more accurately on the complete set of samples. The justification of this can be proved in further experimentation sections of the current chapter.

4.2 Experimentation in Text Vectorization

To begin with, the main reason of performing text vectorization is that raw data (clinical text), a sequence of symbols cannot be input directly into the algorithms themselves since most of them expect numerical feature vectors with a fixed size rather than the raw text documents, which have a variable length.

Therefore, we applied vectorization, which is a general process of turning a collection of text documents into numerical feature vectors. This specific strategy considered, BOW, comprises different pre-processing tasks, as stop words removal, tokenization, counting and normalization.

Documents are described by word occurrences/relevance while completely ignoring the relative position of the words in the document. Furthermore, as word relevance analyzer we considered a *Tfidf-Vectorizer* by resulting the most descriptive indicator.
This indicator can be seen as a*local-global* analysis of each term (lemma).The local indicator can be referred as the first *Tfidf* component *Term Frequency*, which gives an insight of how relevant is the word by its frequency in a particular document.
Besides, global indicator *Tfidf* second component, gives the insight of how discriminant the lemma by the number of documents that contain the term.

1. *TfidfVectorizer* parameters:

   - $stop\_words$
   - max_df
   - min_df
   - $max\_features$
   - $ngram\_range$

- To start experimentations, we considered an *English* $stop\_words$ removal, since these terms do not provide any discriminant information by having a higher frequency among the text.

- In addition, based on *Python Sklearn* technical configurations on *Tfidf Vectorizer*, we describe the min_df and max_df components.

  max_df parameter controls the words considered by text vectorization by limiting words that appear in a certain number of documents. The *Python Sklearn* framework allow to give integer values (i.e. number of documents) or real values (i.e. the fraction of documents) as constrain, therefore if max_df is set on 0.50 it will only considered lemmas that appear in 50% of CORPUS documents approximately.

  Consequently, after some experimentations with the different real values configurations: $[0.0, 1.0]$ in intervals of 0.25, we found that the best value for the attribute max_df is 0.25, based on the framework functionality, min_df is considered by default as 0.0.

- Furthermore, for $max\_features$ parameter we did not set any special value along with $ngram\_range = (1, 1)$ to observe how many different and relevant terms according to the previous vectorizer parameters recently sat. We found $15,296$ as the maximum intrinsic number of features. Additionally, considered the different percentages $[0.125, 0.25, 0.50, 0.75, 1.0]$ % as different $max\_features$ (number of features constraint) configurations.

- Finally, $ngram\_range$ as the last parameter to tune in Text Vectorization, we tried different configurations considering: mono-grams, bi-grams and tri-grams. This assumptions are based on medical text work done from literature [39], in which they suggest the importance of considering the n-gram up to 3. Moreover, vectorizer can handle two types of ngram_ranges, fixed BOW n_gram

length (e.g. $(1,1)$ where BOW has n_grams within same number of words) or variable length (e.g. $(1,3)$ where BOW has n_grams within variable number of words, one, two or three words in a token). We considered the following different *ngram_ranges* pairs of values $[(1,1),(1,2),(1,3),(2,2),(2,3),(3,3)]$.The *ngram_ranges* tuple components represent the minimum and the maximum number of words considered for a given text vectorization.

The following *TextVectorization* experimental results aim to find a leading path of which are the most suitable representations of the data.

This fact considers different configurations *max_features* and *ngram_ranges* in order to find which configurations minimize MSE and $1 - R^2$.

As we previously mentioned in *Chapter 2*, MSE is the traditional metric to asses prediction errors, and $1 - R^2$ describes how great is the $\bar{Y}$ error with relation to the $var(Y)$,(i.e. if the variance of the truth is higher than the prediction error, the model may be induced from the data in a proper way).

We initially considered a KPS prediction without any translation to ECOG scale, a decimal rounding, a 5-10 multiple rounding (considering KPS 10-stage intervals) and a post-prediction maximum values constrain over the $Full - Set$ of samples (4023) CT.
All this considerations were taken into account at implementing PLS and MLP models to this task. Hence we calculated the variance per every KPS score from the truths, $var(y_{min}) = 298.939$ and $var(y_{max}) = 238.065$ for posterior comparison purposes.

After computing all the different combination of *Text Vectorization* configurations we found the following learning behaviors in PLS and MLP simulations respectively:

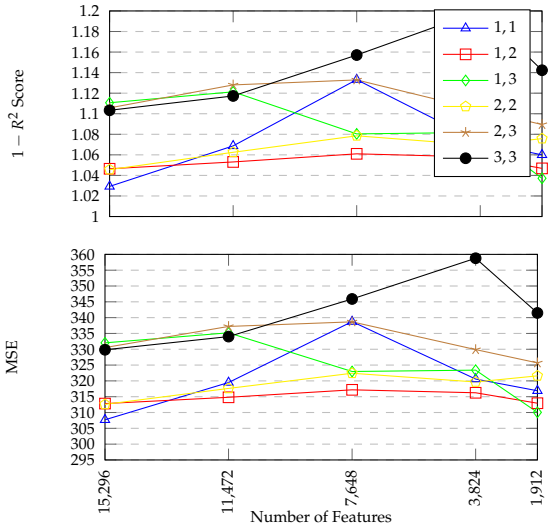

FIGURE 4.1: 10-CV Testing KPS_{min} : $1 - R^2$ & MSE scores obtained by different *ngram_range* & *max_features* configurations implementing PLS.

As we can observe in the preliminary simulations in PLS to predict KPS_{min}, the most suitable *ngram_combinations* are located in middle representations of lemmas as features as (1,2) and (2,2) (i.e. features that represent one or two lemmas, and features that represent two lemmas). For this configurations, MSE and $1 - R^2$ minimizations show a consistent behavior, leading to the minimum values. Furthermore, the worst features representations are (2,3) and (3,3) representing complex features with two or more lemmas contained.

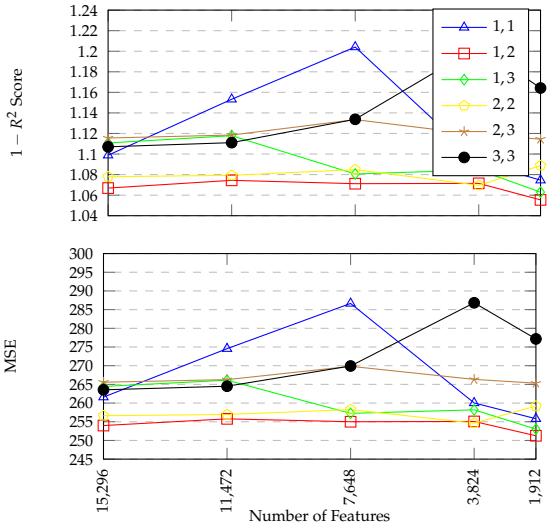

FIGURE 4.2: 10-CV Testing KPS_{max} : $1 - R^2$ & MSE scores obtained by different *ngram_range* & *max_features* configurations implementing PLS.

As we can observe in the preliminary simulations in PLS to predict KPS_{max} are congruent with simulation behaviors of KPS_{min}. The most suitable *ngram_combinations* are located in middle representations of lemmas as features as (1,2) and (2,2) (i.e. features that represent one or two lemmas, and features that represent two lemmas). For this configurations, MSE and $1 - R^2$ minimizations show a consistent behavior, leading to the minimum values. Remarkably, the worst features representations are (1,3) and (3,3) representing configurations from extreme bounds considered. This may suggest that in general features containing three lemmas may be complex and do not provide discriminant information.

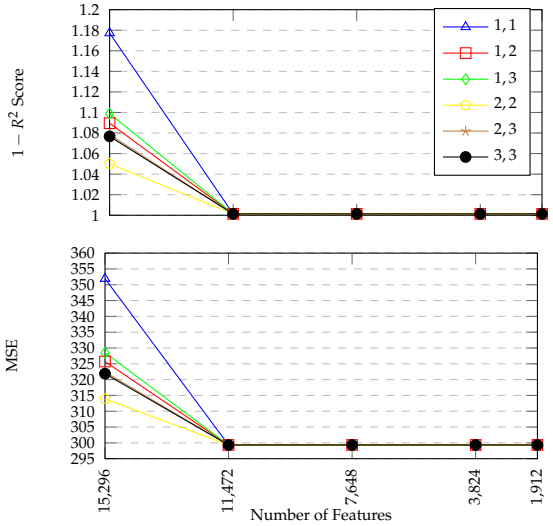

FIGURE 4.3: 10-CV Testing KPS_{min} : $1 - R^2$ & MSE scores obtained by different *ngram_range* & *max_features* configurations implementing MLP.

As we can observe in the preliminary simulations in MLP to predict KPS_{min}, neural network default parameters over fit when the number of features decrease from the maximum. Based only on the results considering the maximum number of features we can observe that the most suitable *ngram_combinations* are located in complex features representations, in which there are more than one lemma contained. For this configurations, MSE and $1 - R^2$ leading to the minimum values in proportion to the number of lemmas considered in the text vectorization. Remarkably, the worst features representations are (1,1) representing low complexity features compositions. This may suggest that MLP learning may generalize better among complex features representations in text vectorization.

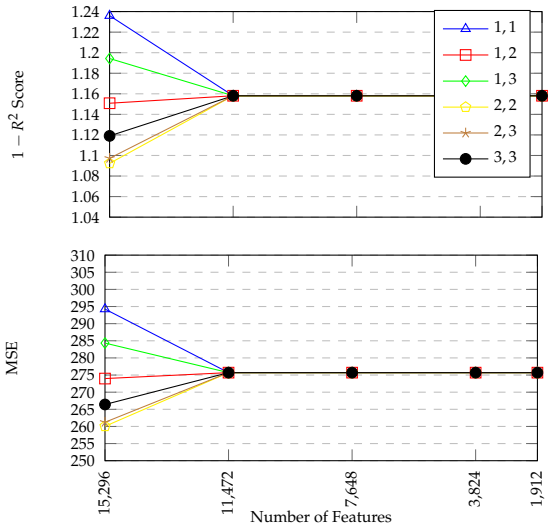

FIGURE 4.4: 10-CV Testing KPS_{max} : $1 - R^2$ & MSE scores obtained by different *ngram_range* & *max_features* configurations implementing MLP.

As we can observe in the preliminary simulations in MLP to predict KPS_{max}, results seem to be congruent with KPS_{max} predictions. As we can see, neural network default parameters over fit when the number of features decrease from the maximum. Based only on the results considering the maximum number of features we can observe that the most suitable *ngram_combinations* are located in complex features representations, in which there are more than one lemma contained in features except from (1,3) configurations. For (1,2), (2,2), (2,3) and (3,3) configurations, MSE and $1 - R^2$ lead to the minimum values. Remarkably, the worst features representations are (1,1) and (1,3) representing low complexity features compositions. This may confirm that MLP learning may generalize better among complex features representations in text vectorization.

After analyzing the behavior of the algorithms, we can infer that MLP with a default configuration has a strong tendency to over fit the data after feature removing compared to PLS model.
Furthermore, the following tables that summarize plots results denote the most profitable data representations per every *ngram_range* configurations according to $1 - R^2$ & MSE scores minimization. This results display the type of *Text Vectorization* that seem to be more suitable to train a prediction model.

$Ngram$	$Features$	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
1,1	15 296	1.029	1.098	307.698	261.595
1,2	1912	1.046	1.055	312.942	251.252
1,3	1912	1.037	1.062	310.053	253.040
2,2	15 296	1.045	1.077	312.574	256.618
2,3	1912	1.089	1.114	325.693	265.326
3,3	15 296	1.103	1.107	329.836	263.544

TABLE 4.3: Best 10-CV testing MSE & $1 - R^2$ configurations obtained for PLS.

$Ngram$	$Features$	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
1,1	11 472	1.001	1.158	299.335	275.702
1,2	11 472	1.001	1.158	299.335	275.702
1,3	11 472	1.001	1.158	299.335	275.702
2,2	15 296	1.050	1.092	313.916	260.040
2,3	11 472	1.001	1.158	299.335	275.702
3,3	11 472	1.001	1.158	299.335	275.702

TABLE 4.4: Best 10-CV testing MSE & $1 - R^2$ configurations obtained for MLP.

We can observe from the results of the *BreastCancer* CT *Text Vectorization* that PLS with default configuration tends to generalize more precisely over simple $ngram_range$

configurations (i.e. in which *BOW* terms contain tokens with one lemma) than complex features representations. Besides, linear models seem to fit better in extreme-size cases, in which data has big or small sizes.

Furthermore, as PLS, *MLP* seem to generalize better under non-complex features representations in *Text Vectorization*.

Interestingly, *MLP* with default parameters seems to over fit quickly and get stuck in local minima at weight adjustment. Therefore, the non-linear algorithm does not improve metrics when the number of features decrease from the maximum value as it can be seen in *Figures 4.3 & 4.4*.

After the *Text Vectorization* analysis, we considered other forms to boost predictive models learning, one of them relies in *Dimensionality Reduction or Feature Selection*. Dimensionality reduction considers reducing number of features to avoid *The Curse of Dimensionality Problem*, in which the data dimensionality, space and sparseness increase proportionally. Consequently, sparsity and higher dimensions represent a learning issue by statistical models at developing a high-variance model, especially in cases, intending to fit higher dissimilarities between samples.

In order to avoid *The Curse of Dimensionality Problem*, we are going to consider *Dimensionality Reduction* by applying a data projections in low-dimensional spaces. In which the available max features considered are going to be combined and represented in a more discriminant form.

4.3 Experimenting in Data Projections

At *Data Projection* experiments we are going to consider the *Single Value Decomposition (SVD)* previously mentioned in *Chapter 2* by the advantages previously described and its general matrix decomposition framework suitable for sparse data.

Ideally, we considered dimensionality reduction intending to keep the most representative data attributes in a lower dimension considering the number of samples. This fact will result in robustness to *The Curse of Dimensionality Problem*. In the following simulations we considered the full set of samples taking into account $var(y_{min}) = 298.939$ and $var(y_{max}) = 238.065$, respective variances of KPS_{min} & KPS_{max} scores.

$Ngram$	$Features$	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
$1,1$	32	0.996	1.051	297.888	250.373
$1,2$	252	0.964	1.025	288.252	244.253
$1,3$	252	0.950	1.022	284.128	243.382
$2,2$	252	0.980	1.015	293.197	241.816
$2,3$	252	0.987	1.017	295.217	242.167
$3,3$	503	0.995	1.017	297.741	242.338

TABLE 4.5: Best 10-CV testing MSE & $1 - R^2$ configurations obtained for PLS for different SVD dimensions.

$Ngram$	$Features$	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
1,1	63	1.018	1.109	304.328	264.063
1,2	63	0.994	1.069	297.294	254.617
1,3	63	0.998	1.080	298.512	257.321
2,2	63	1.008	1.081	301.598	257.544
2,3	252	1.001	1.158	299.335	275.702
3,3	252	1.001	1.158	299.335	275.702

TABLE 4.6: Best 10-CV testing MSE & $1 - R^2$ configurations obtained for MLP for different SVD dimensions.

After results obtained in *Table 4.5* and *Table 4.6*, we observe that after applying SVD projections over *Text Vectorization* data, PLS models seem to be consequent in generalization metrics over simple $ngram_range$ configurations. This fact denotes that PLS best generalization results, comprise BOW's terms contain tokens with single lemmas. Moreover, linear models seem to generalize in a better in SVD projections than only *Text Vectorization* data. Furthermore, for PLS, SVD projections of approximately 250 projected features seem to be representative for generalization purposes.

Additionally, MLP model seem to be consequent to *Text Vectorization* generalization. Furthermore, after data mapping, non-linear models seem to generalize better on reasonably lower SVD dimensions as 63 approximations in number of features.

4.4 Experimenting in Prediction Refining

After a brief review analysis of data representation from *Text Vectorization* and *Data Projection* in PLS and MLP default tuning, we found that both of the models show a better learning performance among data representations that contain mono-lemma terms in BOW.

This fact can be observed particularly at following *ngram_ranges*: 1,1, 1,2, and 1,3. Furthermore, SVD data projections seem to re-fine the learning for each data representation, since it can be seen that best data projections are found approximately in the same *ngram_ranges* as the *Text Vectorization* data representations.

Interestingly, aiming to improve generalization in *Breast Cancer CT* profile predictions, we decide to devote some experimentations related to refining final predictions. We consequently considered that prediction refining may lead to a better scores in testing results.

Therefore, for prediction refining we considered 3 aspects as potential tunings: a scale translation KPS-ECOG in predictions based on equivalences table in *Chapter 1*, a decimal rounding (rounding real numbers to the close integer value), and a tens rounding (rounding integers modules of 10 to the closest tens number, only KPS predictions). Therefore, we considered to overview the following cases of data predictions post-processing.

(A) *KPS* prediction, decimal and non-tens rounding.

(B) *KPS* prediction, decimal and tens rounding.

(C) *ECOG* prediction, decimal and non-tens rounding.

(D) *ECOG* prediction, decimal and tens rounding of *KPS* prediction with posterior translation to ECOG.

Since A and B scenarios consider the same prediction *KPS* scale, we established comparisons among them based on the best experimental results found for all the combinations of features $[15296, 11472, 7648, 3824, 1912]$ within the best BOW representations in *ngram_ranges*: $(1,1)$, $(1,2)$, and $(1,3)$ for both models, *PLS* and *MLP* respectively.

Algorithm	*Ngram*	*Features*	$(A)1-R^2_{min}$	$(B)1-R^2_{min}$	$(A)1-R^2_{max}$	$(B)1-R^2_{max}$
PLS	1,1	15296	1.001	1.029	1.056	1.098
PLS	1,2	1912	1.017	1.046	1.028	1.055
PLS	1,3	1912	1.013	1.037	1.038	1.062
MLP	1,1	11472	1.002	1.001	1.005	1.158
MLP	1,2	11472	1.002	1.001	1.005	1.158
MLP	1,3	11472	1.002	1.001	1.005	1.158

TABLE 4.7: Comparisons among best configurations in 10-CV testing $1-R^2$ at A and B *Text Vectorization* data representations in *PLS* and *MLP*.

Algorithm	*Ngram*	*Features*	$(A)MSE_{min}$	$(B)MSE_{min}$	$(A)MSE_{max}$	$(B)MSE_{max}$
PLS	1,1	15296	299.520	307.698	251.534	261.595
PLS	1,2	1912	304.025	312.942	244.890	251.252
PLS	1,3	1912	302.993	310.053	247.241	253.040
MLP	1,1	11472	299.643	299.335	239.352	275.702
MLP	1,2	11472	299.643	299.335	239.352	275.702
MLP	1,3	11472	299.643	299.335	239.352	275.702

TABLE 4.8: Comparisons among best configurations in 10-CV testing *MSE* at A and B *Text Vectorization* data representations in *PLS* and *MLP*.

After experimenting, we found that the most accurate form to refine predictions in terms of $1-R^2$ and *MSE* scores is A: *KPS* prediction, decimal and non-tens rounding. In order to clarify tens rounding let us denote it as the rounding of modules of 10 integers to the closest tens number.
After that, we can infer that for KPS prediction tens rounding refining is not useful for obtaining more accurate predictions in both linear and non-linear models.

Since *C* and *D* cases consider *ECOG* predictions, we established comparisons among them based on the best experimental results found for all the combinations of features [15296, 11472, 7648, 3824, 1912] within the best BOW representations in *ngram_ranges*: $(1,1)$, $(1,2)$, and $(1,3)$ for both models, *PLS* and *MLP* respectively.

Algorithm	*Ngram*	*Features*	$(C)1-R^2_{min}$	$(D)1-R^2_{min}$	$(C)1-R^2_{max}$	$(D)1-R^2_{max}$
PLS	1,1	1912	1.160	1.169	1.200	1.104
PLS	1,2	15 296	1.122	1.135	1.213	1.107
PLS	1,3	1912	1.131	1.130	1.209	1.126
MLP	1,1	11 472	1.201	1.201	1.117	1.117
MLP	1,2	11 472	1.201	1.201	1.117	1.117
MLP	1,3	11 472	1.201	1.201	1.117	1.117

TABLE 4.9: Comparisons among best configurations in 10-CV testing $1-R^2$ at *C* and *D Text Vectorization* data representations in *PLS* and *MLP*.

Algorithm	*Ngram*	*Features*	$(C)MSE_{min}$	$(D)MSE_{min}$	$(C)MSE_{max}$	$(D)MSE_{max}$
PLS	1,1	1912	0.996	1.004	0.696	0.641
PLS	1,2	15 296	0.964	0.975	0.704	0.643
PLS	1,3	1912	0.971	0.970	0.702	0.653
MLP	1,1	11 472	1.031	1.031	0.648	0.648
MLP	1,2	11 472	1.031	1.031	0.648	0.648
MLP	1,3	11 472	1.031	1.031	0.648	0.648

TABLE 4.10: Comparisons among best configurations in 10-CV testing *MSE* at *C* and *D Text Vectorization* data representations in *PLS* and *MLP*.

After the experiment and a review of the $1-R^2$ & *MSE* scores, we can conclude that *D*, a *KPS* prediction with decimal and tens rounding translated to *ECOG* bring a better learning performance than *C*, a natural *ECOG* prediction without tens rounding.
For this reason, we can infer that predicting *KPS* and a posterior translation to *ECOG* seem to be more feasible in terms of learning than learning *ECOG* responses and predict them. All this seems to be related to the fact that higher range scales overcome in certain form the data response unbalance. In highlights, that in this particular application samples have a unbalanced distribution.

4.5 Experimenting without Problematic Sample Cases

After performing the experiments related to post-processing data predictions, we considered keeping the most profitable approaches to refine predictions, *A* and *D*.

Furthermore, we considered to overview another potential improvement related to generalization of models. We considered a samples splitting according to the compatibility degree criteria in ranges proposed in this work assessing how close is the prediction in terms of ranges of values comparisons. The compatibility criteria assumption is based on the following mathematical expression:

$$CD(y_i, \hat{y_i}) = 1.0 * \frac{length(get\_scales(max(y_{i,min}, y_{i,\hat{min}}), min(y_{i,max}, y_{i,\hat{max}})))}{length(get\_scales(min(y_{i,min}, y_{i,\hat{min}}), max(y_{i,max}, y_{i,\hat{max}})))}$$

where:

i : index of i-th sample.

y_i : i-th response values (min and max).

$\hat{y_i}$: i-th predicted response values (min and max).

$CD()$: compatibility degree function among two ranges.

$length()$: number of stages in the range.

$get_scales()$: generation of range stage list by min and max values.

$max()$: maximum value of a list of numbers.

$min()$: minimum value of a list of numbers.

$y_{i,min}$: i-th (KPS/ECOG) min response value.

$y_{i,max}$: i-th (KPS/ECOG) max response value.

$y_{i,\hat{min}}$: i-th (KPS/ECOG) min response predicted value.

$y_{i,\hat{max}}$: i-th (KPS/ECOG) max response predicted value.

Furthermore, we considered this criteria as an alternative informative metric to assess how close are the predictions to response variables considering both *min* and *max* range of predictions.

Interestingly, after performed a manual *Microsoft Excel* analysis of this compatibility criteria values, we decided to split dataset samples by $CD <= 0.25$ threshold criteria. Consequently, we found that the lower CD split samples that represent the most problematic are featured by lower bounding range response KPS/ECOG values and mono KPS/ECOG values (i.e. ranges where response variables *min*, *max* have the same value).

On the other hand, we found that the higher CD split featured by higher ranges in response comprise 94% of the samples from the $Full - Set$. Therefore, 6% of problematic cases can be discard as not having a statistical relevance from the population of samples.

Consequently we may consider the fact that lower CD samples increase noise at build a general model including the complete set of samples. Based on the previous fact, we put aside the 256 problematic samples to evaluate the $3,767$ samples related to the $CD > 0.25$ trying to observe how does this data sub setting result at the model learning.

The following experiments intend to establish comparisons among the $3,767$ and

4,023 samples sizes considering *A-KPS prediction* and *D-ECOG prediction* previously selected at *Prediction Refining* section. For differentiation purposes in terms of how difficult may result build each model, we must describe the variances that each of the datasets have in *KPS* & *ECOG* scales for *min* & *max* scores in the prediction task.

(I) Complete set of samples (4,023 samples)

Scale	**min**	**max**
KPS	298.939	238.065
ECOG	0.858795	0.580649

TABLE 4.11: Complete set of samples (I): *ECOG* and *KPS* response variables variances.

(II) Non-problematic set of samples (3,767 samples)

Scale	**min**	**max**
KPS	158.615783	47.333426
ECOG	0.407559	0.115017

TABLE 4.12: Non-problematic set of samples (II): *ECOG* and *KPS* response variables variances.

Algorithm	*Ngram*	*Features*	$(I)1-R^2_{min}$	$(II)1-R^2_{min}$	$(I)1-R^2_{max}$	$(II)1-R^2_{max}$
PLS	1,1	15 296	1.001	0.916	1.056	1.016
PLS	1,2	1912	1.017	0.931	1.028	0.984
PLS	1,3	1912	1.013	0.931	1.038	0.989
MLP	1,1	11 472	1.002	1.000	1.005	0.999
MLP	1,2	11 472	1.002	1.000	1.005	0.999
MLP	1,3	11 472	1.002	1.000	1.005	0.999

TABLE 4.13: Best Testing 10-CV $1-R^2$ for *Text Vectorization* without SVD projections for (I) and (II) sets of samples at A (KPS) prediction refine for *PLS* and *MLP*.

Algorithm	*Ngram*	*Features*	$(I)MSE_{min}$	$(II)MSE_{min}$	$(I)MSE_{max}$	$(II)MSE_{max}$
PLS	1, 1	15 296	299.520	145.446	251.534	48.102
PLS	1, 2	1912	304.025	147.693	244.890	46.608
PLS	1, 3	1912	302.993	147.714	247.241	46.849
MLP	1, 1	11 472	299.643	158.618	239.352	47.330
MLP	1, 2	11 472	299.643	158.618	239.352	47.330
MLP	1, 3	11 472	299.643	158.618	239.352	47.330

TABLE 4.14: Best Testing 10-CV *MSE* for *Text Vectorization* without SVD projections for (I) and (II) sets of samples at *A* (KPS) prediction refine for *PLS* and *MLP*.

Algorithm	*Ngram*	*Features*	$(I)1 - R^2_{min}$	$(II)1 - R^2_{min}$	$(I)1 - R^2_{max}$	$(II)1 - R^2_{max}$
PLS	1, 1	1912	1.169	1.356	1.104	1.086
PLS	1, 2	15 296	1.135	1.351	1.107	1.093
PLS	1, 3	1912	1.130	1.408	1.126	1.086
MLP	1, 1	11 472	1.201	1.775	1.117	1.086
MLP	1, 2	11 472	1.201	1.775	1.117	1.086
MLP	1, 3	11 472	1.201	1.775	1.117	1.086

TABLE 4.15: Best Testing 10-CV $1 - R^2$ for *Text Vectorization* without SVD projections for (I) and (II) sets of samples at *D* (ECOG) prediction refine for *PLS* and *MLP*.

Algorithm	*Ngram*	*Features*	$(I)MSE_{min}$	$(II)MSE_{min}$	$(I)MSE_{max}$	$(II)MSE_{max}$
PLS	1, 1	1912	1.004	0.552	0.641	0.125
PLS	1, 2	15 296	0.975	0.550	0.643	0.125
PLS	1, 3	1912	0.970	0.574	0.653	0.125
MLP	1, 1	11 472	1.031	0.723	0.648	0.125
MLP	1, 2	11 472	1.031	0.723	0.648	0.125
MLP	1, 3	11 472	1.031	0.723	0.648	0.125

TABLE 4.16: Best Testing 10-CV *MSE* for *Text Vectorization* without SVD projections for (I) and (II) set of samples at *D* (ECOG) prediction refine for *PLS* and *MLP*.

After performing the experiments, we found that MSE and $1 - R^2$ scores are minimized in the particular scenario of A KPS prediction refining of models learning in set without problematic samples.

Notably, set without problematic samples has a lower variance for *min* and *max* response variables compared to the complete set of samples as it can be seen in *Table 4.11* and *4.12*. Therefore, as it was shown in previous results in order to optimize (maintain or decrease) $1 - R^2$ scores it is required to obtain lower MSE.
This fact suggest that for the particular case of A KPS prediction refining considering a subset of samples without problematic cases result more profitable than using the complete set of samples.

On the other hand, in the particular case of D ECOG prediction refining, we found that for the complete set samples learning MSE are higher than set without problematic samples. Besides, as it was previously seen in *Tables 4.11* and *Tables 4.12* response variable variances in complete set of samples are higher than set of samples without problematic cases. As an additional fact, $1 - R^2$ scores in the complete set are lower than set without problematic samples.

This fact suggests MSE scores related to the set without problematic samples scenario are not low enough in relation with their response variables variances. For this reason, in the particular case of predicting D ECOG prediction refining, a complete set of samples have to be considered.

Moreover, despite the fact that prediction errors MSE in both *Predictions Refining* approaches A and D are not comparable since predictions are in different scales. We found that approach A, which represents a KPS prediction without tens rounding seem to be more profitable than approach D that considers a tens rounding in KPS prediction with a posterior $ECOG$ scale translation in terms minimizing $1 - R^2$.

After analyzing prediction results without problematic cases we decided to keep only the set without problematic samples, which represent the 94% of samples from the entire dataset for predicting KPS.

Furthermore, after experimentation on D ECOG predicting refining, we decided to perform ECOG learning with the complete set of samples. Summing up, we selected the following partitions refining with their most suitable representation of the data for further experiments:

- KPS prediction, decimal and non-tens rounding in set without problematic samples.
- $ECOG$ prediction, decimal and tens rounding of KPS prediction with posterior translation to ECOG in complete set.

After that, going further with the best results obtained with *Text Vectorization* without SVD, we proceed assessing results considering SVD projections and we found the following experimental results:

Algorithm	*Ngram*	*Features*	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
PLS	1, 1	252	0.907	0.976	143.987	46.205
PLS	1, 2	252	0.897	0.962	142.356	45.559
PLS	1, 3	252	0.898	0.969	142.483	45.866
MLP	1, 1	63	0.957	1.055	151.798	49.965
MLP	1, 2	63	0.953	1.108	151.274	52.490
MLP	1, 3	63	0.935	1.037	148.441	49.113

TABLE 4.17: Best Testing 10-CV $1 - R^2$ and MSE for (II) set of samples with *Text Vectorization* and SVD projections in A (KPS) predictions refining for PLS and MLP.

Algorithm	*Ngram*	*Features*	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
PLS	1, 1	252	1.130	1.109	0.971	0.644
PLS	1, 2	503	1.087	1.074	0.934	0.624
PLS	1, 3	252	1.089	1.098	0.935	0.638
MLP	1, 1	63	1.190	1.147	1.022	0.666
MLP	1, 2	63	1.165	1.121	1.001	0.651
MLP	1, 3	63	1.164	1.151	1.000	0.668

TABLE 4.18: Best Testing 10-CV $1 - R^2$ and MSE for (I) set of samples with *Text Vectorization* and SVD projections in D (ECOG) predictions refining for PLS and MLP.

As a removal of problematic samples experiment conclusion, we found that data predictions seem to generalize in more accurate way over KPS scale. Additionally, we can observe that in KPS predictions PLS models generalize better for 252 features size approximation for every ngram_range considered. On the other hand, MLP generalize better under lower features size approximation as 63.

Moreover, this results denote an improvement in comparison with all the results obtained for KPS predictions in *Chpater 5: Simulations*.
This conclusion considers the fact that even if the samples of data are 6% less, and *min* and *max* scores variances are lower, errors have decreased considerably in relation with the lower variance of the response variables from the 94% of the samples, leading to optimum values $1 - R^2$ and MSE found in all the experimentations.
Notably, ECOG prediction MSE require a more deep review based on the fact of lower scale ranges and units in comparison than KPS scale and units. Considering that lower MSE values may not represent good approximations and accurate models in ECOG predictions.

4.6 Model Selection configurations

After multiple considerations related to the data representations, predictions refining, and removing of problematic samples, we proceed trying to find optimum results by selecting the best parametric configuration of the models implemented in

this work. Remarkably, as we observed in previous sections, KPS prediction seem to be a more suitable task based on the results obtained in comparison to ECOG predictions results. Therefore, we are going to proceed with models tuning only considering the KPS prediction with decimal and non-tens rounding with set of samples without problematic cases. Consequently, we are going to leave aside ECOG predictions for future work. In short we are going to overview the final results obtained representing the best configurations found of each model.

4.6.1 Partial Least Squares (PLS)

To remark, experiments related to different PLS configurations are found in *Appendix C.1 Partial Least Squares Results*. These experiments comprise the following parameters:

- PLS number of components: $[1, 2, 3, 4, 5, 10, 20, 50]$.
- PLS normalizations values: $[False, True]$.
- Features representations as $ngram_range$ lemmas: $[(1,1), (1,2), (1,3)]$.
- SVD number of features mappings:
  $[500, 450, 400, 350, 300, 250, 240, 230, 220, 210, 200, 190, 180, 170, 160, 150, 100, 50]$.

After looking at the results in *Appendix C.1*, we found that PLS number of components seem to have an inversely proportional relation with the number of features used. The higher number of components is correlated with lower number of features required.

Furthermore, after performing simulations with different number of components in the PLS algorithm as previously mentioned, we established comparisons among PLS normalization parameter in either $0 : False$ or $1 : True$ values. And we obtained the following concluding results for $1 - R^2$ at min and max KPS predictions.

$PLSComponents$	$ngram\_combination$	$Features$	$Norm = 1$	$Norm = 0$
1	1, 2	240	0.8970	0.9331
2	1, 2	240	0.8874	0.8967
3	1, 3	240	0.8873	0.8834
4	1, 3	200	0.8943	0.8883
5	1, 3	200	0.8927	0.8847
10	1, 3	200	0.8926	0.8900
20	1, 3	200	0.8926	0.8924
50	1, 3	200	0.8924	0.8926

TABLE 4.19: KPS_{min} : $1 - R^2$ for different PLS components, $ngram_range$ and number of features configurations.

$PLSComponents$	$ngram\_combination$	$Features$	$Norm = 1$	$Norm = 0$
1	1,3	400	0.9650	1.0130
2	1,3	350	0.9550	0.9811
3	1,2	250	0.9495	0.9717
4	1,2	250	0.9579	0.9577
5	1,2	250	0.9571	0.9637
10	1,2	250	0.9561	0.9597
20	1,2	250	0.9583	0.9565
50	1,2	240	0.9592	0.9594

TABLE 4.20: KPS_{max} : $1 - R^2$ for different PLS components, $ngram_range$ and number of features configurations.

After all the experimentation, related to PLS tuning on KPS predictions, we can conclude that the best PLS configurations found for this type of prediction are:

1. KPS-PLS $1 - R^2_{min} = 0.8834$
   - $components = 3$
   - $normalization = False$
   - $ngram\_combination = (1, 3)$
   - $SVDfeatures = 240$
2. KPS-PLS $1 - R^2_{max} = 0.9495$
   - $components = 3$
   - $normalization = True$
   - $ngram\_combination = (1, 2)$
   - $SVDfeatures = 250$

Based on the concluding results of *min* and *max* for each different type of prediction *KPS*. We found that each response variable to be predicted *min* and *max* are minimized by the PLS model in different configurations of $ngram_range$ and number of features. Although, the number of components in the PLS algorithm in the best results obtained for each prediction are the same. This may suggest that the task considered seem to be tough to model and that improving results seem to be related with the data representations and considerations than with a bi factorial linear model as PLS.

4.6.2 Multilayer Perceptron (MLP)

In order to reference properly to the experiments related to different MLP configurations, these experimental simulations are found in *Appendix C.2 Multilayer Perceptron Results*. The experiments comprise the following parameters:

- MLP number of neurons: $[2, 5, 7, 10, 15, 25]$.

- MLP number of epochs: $[5, 10, 25, 50, 75, 100]$.
- MLP activation functions: *identity, logistic, tanh, and relu*.
- MLP α value: $1x10^{-3}$.
- MLP solver: *lbfgs*.
- Features representations as $ngram_range$ lemmas: $[(1,1),(1,2),(1,3)]$.
- SVD number of features mappings:
  $[500, 450, 400, 350, 300, 250, 240, 230, 220, 210, 200, 190, 180, 170, 160, 150, 100, 50]$.

After looking at the results in *Appendix C.2*, we found that MLP activations functions show a generalized behavior at predicting KPS_{min}, in which *most of* the neurons configurations require **higher** number of epochs in comparison with neurons configurations at predicting KPS_{max}. Furthermore, particularly, we can observe that in KPS_{max} predictions **lower** number of neurons result in **higher** number of epochs and vice versa. On the other hand, in KPS_{min} in general for all the configurations of neurons it is required a *constant* value of epochs in most of the cases.

Additionally, we devoted some time to describe in some way the fluctuation founded by each MLP activation function at their KPS_{min}, KPS_{max} predictions $1 - R^2$ performance score. Based on *Appendix C.2* tables we obtained the minimum and maximum feature size per tables, obtained the range, as $range = max - min$ and consequently calculating the average of $(1,1),(1,2),(1,3)$ related to each activation function for both of the $1 - R^2_{min}$ and $1 - R^2_{max}$ values.

$\overline{features\_range\_size}$	*identity*	*logistic*	*tanh*	*relu*
$1 - R^2_{min}$	223.3	220.0	243.3	190.0
$1 - R^2_{max}$	206.6	176.6	166.6	276.6

TABLE 4.21: Fluctuation analysis based on $1 - R^2_{min}$ and $1 - R^2_{max}$ $\overline{features\_range\_size}$ of tables shown in *Appendix C.2* for different $ngram_combinations$, $feature\_size$ configuration considering identity, logistic, tanh and relu activation functions.

Furthermore, after a brief analysis of the average features range size, we found that **identity** and **logistic** activation functions have a more consistent behavior among the feature range sizes of both of the scores in comparison to **tanh** and **relu** that have lower feature ranges for one score, and higher for others. Therefore, this may suggest that having less fluctuations **identity** and **logistic** seem to have a slightly more wide and focalized weight optimization space compared to **tanh** and **relu**. Besides, the fact may also suggest that tuning both functions by different parameters may show a consistent path by different combinations of parameters.

In addition, after performing simulations with different number of components in the MLP algorithm as previously mentioned, we established comparisons among MLP parameters for each activation function, in order to find the best configuration of each activation function. Consequently, we obtained the following concluding results for $1 - R^2$ at *min* and *max* KPS predictions.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{min}$
2	100	1, 3	210	0.8985
5	100	1, 2	500	0.9022
7	100	1, 3	220	0.8965
10	100	1, 3	240	0.8903
15	100	1, 3	190	0.8931
25	100	1, 2	500	0.9009

TABLE 4.22: KPS_{min} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *identity* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{max}$
2	50	1, 3	500	0.9794
5	50	1, 2	180	0.9746
7	25	1, 2	190	0.9785
10	100	1, 2	240	0.9668
15	25	1, 2	350	0.9817
25	25	1, 3	180	0.9807

TABLE 4.23: KPS_{max} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *identity* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{min}$
2	75	1, 3	180	0.8951
5	75	1, 2	450	0.9004
7	75	1, 3	190	0.8965
10	75	1, 3	220	0.9040
15	75	1, 3	200	0.9179
25	75	1, 3	200	0.9052

TABLE 4.24: KPS_{min} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *logistic* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{max}$
2	100	1, 2	230	0.9916
5	50	1, 2	100	0.9852
7	50	1, 2	220	0.9689
10	50	1, 2	250	0.9800
15	25	1, 2	210	0.9999
25	25	1, 2	190	0.9842

TABLE 4.25: KPS_{max} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *logistic* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{min}$
2	50	1, 2	220	0.9256
5	50	1, 3	350	0.9324
7	75	1, 3	100	0.9726
10	50	1, 2	220	0.9235
15	50	1, 2	210	0.9226
25	10	1, 2	230	1.0000

TABLE 4.26: KPS_{min} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *tanh* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{max}$
2	50	1, 3	250	0.9879
5	10	1, 1	350	0.9981
7	50	1, 2	210	0.9965
10	10	1, 2	230	0.9845
15	10	1, 2	210	0.9945
25	10	1, 2	200	0.9999

TABLE 4.27: KPS_{max} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *tanh* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{min}$
2	100	1,2	350	0.9056
5	100	1,2	400	0.8944
7	100	1,3	250	0.9013
10	100	1,3	230	0.9010
15	100	1,3	190	0.8934
25	100	1,2	210	0.9008

TABLE 4.28: KPS_{min} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *relu* activation function in different $ngram_range$ and number of features combinations.

Neurons	*Epochs*	*ngram_combination*	*Features*	$1 - R^2_{max}$
2	25	1,2	240	0.9906
5	25	1,1	500	0.9844
7	25	1,3	200	0.9654
10	25	1,2	400	0.9770
15	25	1,2	100	0.9855
25	100	1,2	250	0.9710

TABLE 4.29: KPS_{max} : $1 - R^2$ for best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations with *relu* activation function in different $ngram_range$ and number of features combinations.

After all the experimentation, related to MLP tuning on KPS predictions, we can conclude that the best MLP configurations found for this type of prediction are:

$1 - R^2$ **Score**	*identity*	*logistic*	*tanh*	*relu*
KPS_{min}	0.8903	0.8951	0.9226	0.8934
KPS_{max}	0.9668	0.9689	0.9845	0.9654

TABLE 4.30: Best Testing 10-CV $1 - R^2$ from all configurations considered in *identity, logistic, tanh, and relu* in *MLP* activation functions.

Notably, considering that the *identity* MLP activation function mimics the same behavior as *Multivariate Linear Regression*, a linear method for inducing data, we must conclude that the best non-linear configuration of the MLP comes with *logistic* activation function based on the results shown in Table 4.30, and in the consistency behavior among the KPS $1 - R^2_{min}$ and $1 - R^2_{max}$ feature range size analysis in Table 4.21.

For a informative description of the MLP best parameters, the best MLP parameters found are:

1. KPS-PLS $1 - R^2_{min} = 0.8951$
   - $activation\_function = logistic$

- $solver = lbfgs$
- $\alpha = 1x10^{-3}$
- $neurons = 2$
- $epochs = 75$
- $ngram\_combination = (1, 3)$
- $SVDfeatures = 180$

2. KPS-PLS $1 - R^2_{max} = 0.9689$
   - $activation\_function = logistic$
   - $solver = lbfgs$
   - $\alpha = 1x10^{-3}$
   - $neurons = 7$
   - $epochs = 50$
   - $ngram\_combination = (1, 2)$
   - $SVDfeatures = 220$

Chapter 5

Closing Results

In this section we are going to describe in a brief way the considerations that were done in order to find the final and conclusive results show further in the current chapter. As it was seen in *Section 4.2: Text Vectorization* for this particular *BreastCancer* CTCP task we conclude that simple $ngram_range$ configurations in text vectorization lead to minimize $1 - R^2$ and MSE scores. This fact considers features representations as combinations of one lemma with two, or three lemmas.

Furthermore, as it was mentioned in *Section 4.3: Data Projections* applying SVD projections over data vectorization, seem to be consequent with metrics minimization over simple $ngram_range$ configurations. This fact enforces the idea of simple $ngram_range$ configurations $(1,1), (1,2), (1,3)$ as a path to follow to optimize learning predictions. Additionally, we conclude that both models seem to generalize better in SVD projections with approximately 250 mapped features.

Moreover, as it was seen in *Section 4.4: Prediction Refining*, we found that the most accurate form to refine predictions in terms of minimizing $1 - R^2$ and MSE scores is considering a KPS prediction with decimal rounding and without tens rounding (rounding modules of 10 values to the closest 10 multiple). Notably, we found that predicting ECOG scores may result hard for inducing the models based on their smaller scale range of values and the unbalanced dataset considered in this application. For this reason, we do not extend any closing result for ECOG scale, and resulting predictions closing metrics are related to KPS scale.

Additionally, as it was mentioned in *Section 4.5: Experimenting without Problematic Sample Cases*, removal of problematic samples from dataset showed that models seem to generalize in more accurate way, particularly, on KPS predictions, showing an improvement in $1 - R^2$ and MSE scores minimization in comparison with the results obtained in previous stages of the experimentation.

We found that problematic samples related to CT have a proper and accurate length of text but comprise cases in which, clinical trial KPS profile is featured by lower ranges (i.e. in which often the maximum value of the range is lower than 60 according to KPS scale) and cases in which prediction is related to a single value (i.e. in which minimum and maximum values of the KPS range comprise the same value). For the sake of a clarifying the problematic samples, response variables min, and max of problematic samples are found in *Appendix D*.

To sum up, this conclusion suggests that removed samples of data consisting in 6% are not statistically relevant in comparison with the 94%. Besides, we found that even when considering 94% of the samples comprise a min and max response scores

lower variance, the errors have decreased considerably in relation with the variance of the response variables leading to the minimal $1 - R^2$ found in all the previous experimentations.

Finally, as we conclude in *Section 4.6: Model Selection* after experimenting with different configurations among the linear and non-linear models, we obtained the following closing results. After all the experimentation on linear models related to PLS tuning on KPS predictions, we can conclude that the best PLS configurations found for this type of prediction are:

1. KPS-PLS $1 - R^2_{min} = 0.8834$
    - $components = 3$
    - $normalization = False$
    - $ngram\_combination = (1, 3)$
    - $SVDfeatures = 240$
2. KPS-PLS $1 - R^2_{max} = 0.9495$
    - $components = 3$
    - $normalization = True$
    - $ngram\_combination = (1, 2)$
    - $SVDfeatures = 250$

Moreover, after all the experimentation done with non-linear models, must conclude that the best configuration of the MLP comes with *logistic* activation function based on the results shown in Table 4.30, and in the consistency behavior among the KPS $1 - R^2_{min}$ and $1 - R^2_{max}$ feature range size analysis in Table 4.21.
For a informative description of the MLP best parameters, the best MLP parameters found are:

1. KPS-PLS $1 - R^2_{min} = 0.8951$
    - $activation\_function = logistic$
    - $solver = lbfgs$
    - $\alpha = 1x10^{-3}$
    - $neurons = 2$
    - $epochs = 75$
    - $ngram\_combination = (1, 3)$
    - $SVDfeatures = 180$
2. KPS-PLS $1 - R^2_{max} = 0.9689$
    - $activation\_function = logistic$
    - $solver = lbfgs$
    - $\alpha = 1x10^{-3}$
    - $neurons = 7$
    - $epochs = 50$

- $ngram\_combination = (1, 2)$
- $SVDfeatures = 220$

Furthermore, for effects of comparison between the learning and the testing learning performance we describe the final outcomes of the simulations of PLS and MLP. We take into account as it was previously mentioned in *Chapter 4: Experimentation*, that in every model best configuration of prediction related to *min* and *max* comprise different configurations of the model for both of the response variables predictions. Finally let us establish a learning comparison in terms of $1 - R^2$ and MSE scores from *10 Fold Cross Validation* training and testing in both models.

learning_task	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
Training	0.7766	0.8123	123.1852	38.4503
Testing	0.8834	0.9495	140.1356	44.9453

TABLE 5.1: Learning comparison in terms of $1 - R^2$ and MSE scores from *10 Fold Cross Validation* training and testing in PLS best configurations.

learning_task	$1 - R^2_{min}$	$1 - R^2_{max}$	MSE_{min}	MSE_{max}
Training	0.798 150	0.937 983	126.599 226	44.397 935
Testing	0.895 134	0.968 871	141.982 402	45.859 968

TABLE 5.2: Learning comparison in terms of $1 - R^2$ and MSE scores from *10 Fold Cross Validation* training and testing in MLP best configurations.

Chapter 6

Conclusions

As we can observe from all the researching, experimentation and work done in this thesis, *The Breast Cancer Clinical Trial ECOG-Classification Problem* has many relevant aspects that have to be considered in order to find a proper representation of the data to train a model.

We can start describing the data used in the experiments as small source of clinical trials for breast cancer, we found that now a days the number of breast cancer clinical trials around the world is relatively smaller, comprising less than ten thousand samples. Most of the samples, do not have the relevant information in terms of ECOG or KPS scales scores to be considered in learning experiments, this fact make the classification learning task a hard to model task by insufficient amount of data.

Furthermore, after gathering the samples and keeping only the ones that have PS scale scores as ECOG, KPS or others, we found that the range of values considered as response variables, were unbalanced in terms of minimum, maximum or range in ECOG or KPS scorings. After this, we decided to move forward to a prediction representation of the task and rely on the natural numerical property of the values in PS scales, keeping only the maximum and minimum score and turn the problem into a multivariate regression task.

Furthermore, besides the data considerations and the approaching solution, we can observe from the work done, that *Natural Language Processing* has considerable importance by the fact as it was shown at the experimental results a proper representation of the features as combinations of lemmas, and an appropriate feature size are highly correlated with a discriminant representation of the data. For this particular instance of breast cancer clinical trial analysis, we found that a proper representation of the features involves stop words removal removing non-relevant terms, lemmatization finding the root form of English medical terms, and text vectorization representing data features characterized by relatively simple combinations of lemmas, containing combinations of 1 lemma with 2 or 3 lemmas. Besides, another relevant characteristic related to data representation is feature size that after all the experimentation we found that a useful approximation of numbers of features is given by 250 attributes in data, that in our case are represented as meta-features. These meta-features are projections of the original features for compressing the data dimensions and keeping most of the data variance applying SVD feature selection frameworks. Finally, there were additional conditions considered as prediction refining and removal of problematic cases that increase the learning performance in the experiments.

In addition, related to *Machine Learning* algorithms, we can observe that models are

sensible to their parameters combination, in particular *Multilayer perceptron (MLP)* that comprises more parameters compared to *Partial Least Squares (PLS)*. Besides, we can conclude from the experimental results obtained that models behaviors and best results are congruent approximately with the initial configurations from the beginning of the experimentations. This may suggest that data representations may have more relevance than the algorithms tuning. Besides, we can observe that models behavior do not easily generalize well on both KPS scores using the same data considerations. Every model has slightly different data representations configurations to achieve better results at *min* and *max* scores in predictions. This fact suggests that the multivariate prediction task has certain difficulty in terms of predicting results of two variables with considerably different variances.

Finally we can observed from the closing experimental results that both models, achieve reasonable good scores in $1 - R^2$ and MSE metrics, after establishing comparisons with among *Training* and *Testing* results obtained. These results denoted that scores in metrics are relatively close in both learning simulations, this may suggest that the *Testing* results are approximately close to the optimal results that can be found for this particular task with the conditions previously mentioned. Besides, we can conclude that for this particular type of problem related to *Breast Cancer CT*, and by all the technical and theoretical specifications done, linear models as PLS have an simple and accurate learning performance than non-linear models as MLP.

Appendix A

Clinical Trial Examples

In this section we show two different types of breast cancer CT XML files: the ones with explicit ECOG/KPS score and the ones without ECOG/KPS score. In order to clarify, the XML highlighted fields with scores represent the medical text used in the experiments.

A.1 Example of Explicit ECOG Score in Clinical Trial

```
<?xml version="1.0" encoding="UTF-8"?>
<?
This xml conforms to an XML Schema at: https://clinicaltrials.gov/ct2/
    html/images/info/public.xsd and an XML DTD at: https://
    clinicaltrials.gov/ct2/html/images/info/public.dtd
?>


<clinical_study rank="2051" >

<required_header>
        <download_date>
                ClinicalTrials.gov processed on November 16, 2016
        </download_date>
        <link_text>
                Link to the current ClinicalTrials.gov record.
        </link_text>
        <url>https://clinicaltrials.gov/show/NCT00188604</url>
</required_header>

<id_info>
        <org_study_id>UHN REB 03-0741-C</org_study_id>
        <nct_id>NCT00188604</nct_id>
</id_info>

<brief_title>
        The Use of Selenium to Treat Secondary Lymphedema - Breast
            Cancer
</brief_title>

<official_title>
```

```
        A Randomized Phase II Placebo-controlled Double Blind Study of
            Using Selenium in the Treatment of Secondary Lymphedema in
            Breast Cancer Patients
</official_title>

<sponsors>
        <lead_sponsor>
                <agency>University Health Network, Toronto</agency>
                <agency_class>Other</agency_class>
        </lead_sponsor>
        <collaborator>
                <agency>Princess Margaret Hospital, Canada</agency>
                <agency_class>Other</agency_class>
        </collaborator>
</sponsors>

<source>University Health Network, Toronto</source>

<oversight_info>
        <authority>Canada: Ethics Review Committee</authority>
</oversight_info>

<brief_summary>
        <textblock>
                The primary objective of this study to assess the
                    effectiveness of selenium compared to
                placebo in reducing the lymphedema in-patients with
                    breast cancer. Secondary objectives are
                to assess the impact of selenium on patient's quality of
                    life and to assess the incidence of
                adverse effects of selenium therapy.
        </textblock>
</brief_summary>

<overall_status>Completed</overall_status>

<start_date>January 2004</start_date>

<completion_date type="Actual">January 2009</completion_date>

<primary_completion_date type="Actual">
        January 2009
</primary_completion_date>

<phase>Phase 2</phase>

<study_type>Interventional</study_type>

<study_design>
```

```
        Allocation: Randomized, Endpoint Classification: Safety/Efficacy
             Study, Intervention Model: Crossover Assignment, Masking:
            Double-Blind, Primary Purpose: Treatment
</study_design>

<primary_outcome>
        <measure>
                To assess the effectiveness of orally administered
                    selenium compared to placebo in reducing   arm
                    lymphedema in patients treated with surgery (axillary
                     nodal dissection) and radiotherapy for breast cancer
                    .
        </measure>
</primary_outcome>

<secondary_outcome>
        <measure>To assess the toxicity of selenium.</measure>
</secondary_outcome>

<secondary_outcome>
        <measure>
                To assess the association of selenium, quality of life
                    and limb function.
        </measure>
</secondary_outcome>

<enrollment type="Anticipated">34</enrollment>

<condition>Breast Neoplasms</condition>

<condition>Lymphedema</condition>

<intervention>
        <intervention_type>Drug</intervention_type>
        <intervention_name>sodium selenite</intervention_name>
</intervention>

< eligibility >
        <  criteria  >
                <  textblock >
                        - Patients with clinically documented lymphedema
                            of upper limb secondary to breast  cancer
                            management (surgery - axillary nodal
                            dissection, and radiotherapy)

                        - Patients who have had other modalities of
                            management can be included, e.g. physical
                            therapy, pharmacological therapy

                        -  ECOG performance 0-2
```

```
                    - Informed consent

                    Exclusion Criteria:

                    - Active cellulitis/skin infection of the limb

                    - Venous thrombosis of the upper limbs

                    - Active malignancy

                    - Any other medical condition or congenital or
                      traumatic injury involving either limb

                    - Patients already on selenium medication

                    - Patients participating in another clinical
                      study related to lymphedema
            </ textblock>
        </ criteria >
        <gender>Female</gender>
        <minimum_age>18 Years</minimum_age>
        <maximum_age>N/A</maximum_age>
        <healthy_volunteers>No</healthy_volunteers>
</ eligibility >

<overall_official>
        <last_name>Wilfred Levin, MD</last_name>
        <role>Principal Investigator</role>
        <affiliation>Princess Margaret Hospital, Canada</affiliation>
</overall_official>

<location>
        <facility>
                <name>Princess Margaret Hospital</name>
                <address>
                        <city>Toronto</city>
                        <state>Ontario</state>
                        <zip>M5G 2M9</zip>
                        <country>Canada</country>
                </address>
        </facility>
</location>

<location_countries>
        <country>Canada</country>
</location_countries>

<verification_date>August 2010</verification_date>
```

```
<lastchanged_date>August 12, 2010</lastchanged_date>

<firstreceived_date>September 12, 2005</firstreceived_date>

<has_expanded_access>No</has_expanded_access>

<condition_browse>
        <?
                CAUTION: The following MeSH terms are assigned with an
                    imperfect algorithm
         ?>
        <mesh_term>Breast Neoplasms</mesh_term>
        <mesh_term>Lymphedema</mesh_term>
</condition_browse>

<intervention_browse>
        <?
                CAUTION: The following MeSH terms are assigned with an
                    imperfect algorithm
        ?>
        <mesh_term>Selenious Acid</mesh_term>
        <mesh_term>Sodium Selenite</mesh_term>
        <mesh_term>Selenium</mesh_term>
</intervention_browse>

<? Results have not yet been posted for this study ?>

</clinical_study>
```

A.2 Example of Non-ECOG Score in Clinical Trial

```
<?xml version="1.0" encoding="UTF-8"?>
<?
This xml conforms to an XML Schema at: https://clinicaltrials.gov/ct2/
    html/images/info/public.xsd and an XML DTD at: https://
    clinicaltrials.gov/ct2/html/images/info/public.dtd
?>

<clinical_study rank="5360">

<required_header>
        <download_date>
                ClinicalTrials.gov processed on November 16, 2016
        </download_date>
        <link_text>
                Link to the current ClinicalTrials.gov record.
        </link_text>
        <url>https://clinicaltrials.gov/show/NCT00558168</url>
</required_header>
```

```
<id_info>
        <org_study_id>ARI-IPEP-0104</org_study_id>
        <nct_id>NCT00558168</nct_id>
</id_info>

<brief_title>
        Electronic Study for Anastrozole Pharmacovigilance Evaluation
</brief_title>

<acronym>E-SAFE</acronym>

<official_title>
        Electronic Study for Anastrozole Pharmacovigilance Evaluation
</official_title>

<sponsors>
        <lead_sponsor>
                <agency>AstraZeneca</agency>
                <agency_class>Industry</agency_class>
        </lead_sponsor>
</sponsors>

<source>AstraZeneca</source>
<oversight_info>
        <authority>
                Turkey: Turkish Republic Ministry of Health
        </authority>
        <has_dmc>No</has_dmc>
</oversight_info>

<brief_summary>
        <textblock>
                Collecting information regarding adverse events from
                    patients on treatment with anastrazole
with early stage breast cancer
        </textblock>
</brief_summary>

<overall_status>Completed</overall_status>

<start_date>January 2004</start_date>

<completion_date type="Actual">July 2008</completion_date>

<phase>N/A</phase>

<study_type>Observational</study_type>

<study_design>Time Perspective: Prospective</study_design>

<enrollment type="Actual">1850</enrollment>
```

```
<condition>Early Breast Cancer</condition>

<  eligibility >
       <  criteria >
              <  textblock >
                     Inclusion Criteria:

                     - Post-menopausal Early Invasive Breast Cancer
                        Patients who are under anastrazole treatment,
                        who have normal renal and hepatic functions.

                     Exclusion Criteria:

                     - Metastatic breast cancer patients, previous
                        hormonal therapy, other malignancies.
              </  textblock >
       </  criteria >
       <gender>Female</gender>
       <minimum_age>N/A</minimum_age>
       <maximum_age>N/A</maximum_age>
       <healthy_volunteers>No</healthy_volunteers>
</  eligibility >

<overall_official>
       <last_name>Nejdet Uskent</last_name>
       <role>Principal Investigator</role>
       <affiliation>Kadir Has University Medical School</affiliation>
</overall_official>

<verification_date>July 2008</verification_date>

<lastchanged_date>July 25, 2008</lastchanged_date>

<firstreceived_date>November 13, 2007</firstreceived_date>

<keyword>Anastrazole</keyword>

<keyword>safety</keyword>

<keyword>early breast cancer</keyword>

<has_expanded_access>No</has_expanded_access>

<intervention_browse>
       <? CAUTION: The following MeSH terms are assigned with an
          imperfect algorithm ?>
       <mesh_term>Anastrozole</mesh_term>
</intervention_browse>
```

```
<? Results have not yet been posted for this study ?>

</clinical_study>
```

Appendix B

Bag of Words Examples

B.1 BOW's key features in ngram_range combinations

In this Appendix we are going to show the 25 most discriminant features (words) from every ngram_range combination BOW considered in *Chapter 4*: *Text Vectorization* Experiments. Terms are ordered by $tf - idf$ score metric, which represent the maximum metric value of the term from the $4,023$ complete set of samples.

ngram_range=$(1,1)$		**ngram_range**=$(1,2)$	
Term	**tf-idf**	**Term**	**tf-idf**
old	1.0	hair	0.817760781
swogs	0.977048527	part	0.81684503
specified	0.925059629	soy	0.8108454
part	0.916463409	sibling	0.804823661
phase	0.908855313	step	0.766687391
spouse	0.895830064	phase	0.758143344
participant	0.890146639	prostatectomy	0.745085333
diabetic	0.88544224	fdr	0.73625407
step	0.874052591	nail	0.734587047
subject	0.870429308	hlaa	0.718011719
hair	0.80748843	participant	0.705056657
prostatectomy	0.805428139	swogs swogs	0.703875458
discontinuity	0.802834255	swogs	0.690414709
sonidegib	0.796083969	sonidegib	0.689348137
dca	0.786138212	inguinal	0.688492086
ulrr	0.779853429	gvhd	0.685998585
extranodal	0.777484271	reregistration	0.681479593
bwarm	0.772618304	psa	0.67637177
reregistration	0.770326621	arm	0.676176098
hlaa	0.767640101	subject ha	0.654713585
arm	0.767026005	ret	0.653799487
creat	0.765134332	specified	0.651336634
psa	0.763920221	cmv	0.642204536
mvabn	0.75910658	ulrr	0.63450361
normality	0.758882758	record	0.62805934

TABLE B.1: The most $tf - idf$ discriminant features from *ngram_range* $(1,1)$ and $(1,2)$ *Text Vectorization*.

ngram_range=(1,3)		**ngram_range**=(2,2)	
Term	**tf-idf**	**Term**	**tf-idf**
swogs	0.974902643	swogs swogs	0.994764112
hair	0.859718529	closed accrual	0.914707523
soy	0.810719505	subprotocol aim	0.872349524
part	0.791224417	oral vinorelbine	0.849146378
sibling	0.789594504	subject ha	0.803891806
step	0.780165803	participant ha	0.739818185
prostatectomy	0.752568507	physical activity	0.691454597
hlaa	0.727335025	participant must	0.681512334
participant	0.70793144	subject must	0.671062787
psa	0.706782034	since therapy	0.654498209
phase	0.698159361	patient ha	0.650439363
fdr	0.696680075	targeted tumor	0.637121131
subject ha	0.663776076	step patient	0.636834831
reregistration	0.660796746	epirubicin cyclophosphamide	0.635517485
arm	0.655003728	prior step	0.628799564
record	0.623042586	cancer eligible	0.625840882
octreotide	0.62204368	day cycle	0.624712314
yoga	0.60283539	targeting agent	0.622847026
ulrr	0.598120721	value upper	0.620934334
vinorelbine	0.589652565	step registration	0.613580576
sbrt	0.587545215	dose mvabn	0.611742069
specified	0.585900563	time unv	0.600528281
closed accrual	0.585666175	somatostatin analogue	0.597229692
participant ha	0.583020742	stage metastatic	0.591168519
lenalidomide	0.581676384	subject may	0.58667252

TABLE B.2: The most $tf - idf$ discriminant features from *ngram_range* $(1,3)$ and $(2,2)$ *Text Vectorization*.

ngram_range=(2,3)		**ngram_range**=(3,3)	
Term	**tf-idf**	**Term**	**tf-idf**
swogs swogs	0.996119143	age ecog performance	1.0
closed accrual	0.900508315	carcinoma head neck	1.0
subprotocol aim	0.84260866	swogs swogs swogs	0.995842077
subject ha	0.806135752	registration within wks	0.78140663
participant ha	0.752935778	disease requiring treatment	0.755500359
physical activity	0.749988437	see disease characteristic	0.750993309
epirubicin cyclophosphamide	0.716818304	start treatment day	0.742403015
targeted tumor	0.686087246	first line treatment	0.734258754
participant must	0.675858069	prior step registration	0.729432771
step patient	0.661033133	first day treatment	0.727607422
medical treatment	0.649489722	prior rmpdla injection	0.716119419
dose mvabn	0.635051663	value upper normal	0.710998888
subject must	0.627443852	line metastatic disease	0.707358049
stage metastatic	0.626074874	drink per day	0.702841002
patient ha	0.603925593	infiltrating ductal carcinoma	0.702115072
day cycle	0.595454951	candidate receive therapy	0.697875797
cancer biopsy	0.583260015	criterion patient breast	0.691854846
enrolled arm	0.577880944	providing informed consent	0.686934287
standard treatment	0.573025084	prostate cancer patient	0.682326143
cancer eligible	0.561893099	uln reference lab	0.681195945
ineligible patient	0.548520723	chest wall breast	0.674908719
every week	0.548428036	start protocol therapy	0.670404778
arm patient	0.548387578	karnofsky performance scale	0.669953553
preoperative chemotherapy	0.545958711	exclusion criterion significant	0.667812967
subject may	0.545662684	previous chemotherapy treatment	0.665805647

TABLE B.3: The most $tf - idf$ discriminant features from *ngram_range* $(2,3)$ and $(3,3)$ *Text Vectorization*.

Appendix C

Results in Models Tuning

C.1 Partial Least Squares Results

In this Appendix we are going to show the results obtained by simulating the ML algorithms with different parametric configurations.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9231	0.9133	0.9233	0.9363	0.9419	0.9351	0.9344	0.9352
350	0.9224	0.9097	0.9168	0.9308	0.9318	0.9192	0.9194	0.9217
300	0.9238	0.9144	0.9178	0.9330	0.9319	0.918	0.9185	0.9203
250	0.9141	0.9051	0.9081	0.9185	0.9174	0.9086	0.9088	0.9086
240	0.9178	0.9078	0.9082	0.9170	0.9151	0.9057	0.9054	0.9047
230	0.9172	0.9078	0.9114	0.9221	0.9192	0.9111	0.9112	0.9115
220	0.9168	0.9056	0.9092	0.9170	0.9156	0.9078	0.9077	0.9085
210	0.9120	0.9062	0.9079	0.9139	0.9157	0.9070	0.9070	0.9069
200	0.9151	0.9055	0.9071	0.9137	0.9132	0.9059	0.9061	0.9060
190	0.9116	0.9021	0.9020	0.9098	0.9075	0.9007	0.9008	0.9010
180	0.9103	0.9001	0.9018	0.9017	0.9024	0.8973	0.8974	0.8964
170	0.9125	0.9040	0.9069	0.9104	0.9117	0.9047	0.9048	0.9030
160	0.9175	0.9061	0.9097	0.9123	0.9118	0.9076	0.9077	0.9064
150	0.9238	0.9155	0.9178	0.9235	0.9214	0.9160	0.9158	0.9168
100	0.9294	0.9224	0.9217	0.9243	0.9216	0.9194	0.9196	0.9196
50	0.9390	0.9335	0.9326	0.9341	0.9322	0.9319	0.9319	0.9316
Min	0.9103	0.9001	0.9018	0.9017	0.9024	0.8973	0.8974	0.8964

TABLE C.1: $KPS_{min} : 1 - R^2$ for different PLS components configurations in 1_1 ngram_combination.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9222	0.9113	0.9169	0.9393	0.9404	0.9485	0.9501	0.9501
350	0.914	0.8971	0.9015	0.9156	0.9158	0.9213	0.9222	0.922
300	0.908	0.8954	0.8969	0.9113	0.9112	0.9139	0.9154	0.9154
250	0.9102	0.8988	0.8998	0.9101	0.912	0.915	0.9158	0.9156
240	0.897	0.8874	0.8874	0.8982	0.8973	0.8998	0.9012	0.9041
230	0.9012	0.8903	0.8888	0.8983	0.8977	0.8999	0.9004	0.9018
220	0.9053	0.8946	0.8936	0.9017	0.9004	0.9013	0.9009	0.9012
210	0.9068	0.8996	0.8993	0.9119	0.9123	0.909	0.9095	0.911
200	0.9115	0.9005	0.8991	0.9057	0.9065	0.9065	0.9075	0.9065
190	0.9093	0.8984	0.8977	0.9044	0.9084	0.9073	0.9082	0.909
180	0.905	0.8955	0.8932	0.9002	0.9019	0.901	0.9017	0.9018
170	0.9092	0.8993	0.8988	0.9064	0.9073	0.9043	0.9043	0.9058
160	0.9069	0.8976	0.8973	0.9019	0.9036	0.9028	0.9032	0.9042
150	0.922	0.9127	0.9112	0.9208	0.9216	0.9216	0.9231	0.9219
100	0.9207	0.9112	0.9095	0.9136	0.9139	0.9112	0.9113	0.9132
40	0.9421	0.9339	0.9331	0.9335	0.9359	0.9359	0.9359	0.9358
Min	0.897	0.8874	0.8874	0.8982	0.8973	0.8998	0.9004	0.9012

TABLE C.2: KPS_{min} : $1 - R^2$ for different PLS components configurations in 1_2 ngram_combination.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9233	0.9159	0.9204	0.9446	0.9455	0.9576	0.958	0.958
350	0.9155	0.9033	0.9064	0.921	0.9214	0.9287	0.9278	0.928
300	0.9126	0.9016	0.9017	0.9149	0.9136	0.9176	0.9189	0.9208
250	0.9085	0.8984	0.8975	0.9057	0.9048	0.9075	0.9082	0.9066
240	0.8991	0.891	0.8873	0.8964	0.8954	0.8982	0.8985	0.8992
230	0.9067	0.8992	0.896	0.9057	0.9058	0.908	0.9072	0.9092
220	0.9099	0.9008	0.8972	0.9037	0.903	0.9023	0.9022	0.9021
210	0.9048	0.8953	0.8936	0.9012	0.9009	0.8983	0.8979	0.8988
200	0.9023	0.8898	0.8884	0.8943	0.8927	0.8926	0.8926	0.8924
190	0.9031	0.8944	0.8929	0.8995	0.8987	0.897	0.8973	0.8989
180	0.9038	0.8945	0.8914	0.8973	0.8975	0.8937	0.8937	0.894
170	0.9079	0.8965	0.8949	0.9007	0.9003	0.9	0.8999	0.8992
160	0.9062	0.8973	0.8965	0.9012	0.9014	0.9025	0.902	0.9022
150	0.912	0.9043	0.901	0.908	0.9086	0.9085	0.9094	0.9097
100	0.9258	0.9165	0.9155	0.9178	0.9176	0.9151	0.914	0.9148
50	0.9487	0.9373	0.9353	0.9359	0.9368	0.935	0.9351	0.939
Min	0.8991	0.8898	0.8873	0.8943	0.8927	0.8926	0.8926	0.8924

TABLE C.3: KPS_{min} : $1 - R^2$ for different PLS components configurations in 1_3 ngram_combination.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9836	0.9878	0.9825	0.9962	0.9990	0.9999	1.0005	1.0034
350	0.9847	0.9824	0.9824	0.9875	0.9942	0.9963	0.9970	0.9960
300	0.9876	0.9874	0.9876	0.9970	0.9977	1.0020	1.0028	1.0034
250	0.9777	0.9746	0.9703	0.9805	0.9852	0.9863	0.9863	0.9876
240	0.9842	0.9759	0.9759	0.9810	0.9820	0.9853	0.9854	0.9865
230	0.9889	0.9844	0.9828	0.9908	0.9916	0.9941	0.9946	0.9944
220	0.9853	0.9790	0.9794	0.9872	0.9862	0.9922	0.9930	0.9924
210	0.9863	0.9846	0.9796	0.9814	0.9889	0.9916	0.9917	0.9902
200	0.9881	0.9879	0.9837	0.9868	0.9920	0.9960	0.9962	0.9986
190	0.9844	0.9760	0.9767	0.9778	0.9855	0.9863	0.9865	0.9870
180	0.9885	0.9823	0.9791	0.9808	0.9895	0.9917	0.9921	0.9920
170	0.9916	0.9870	0.9845	0.9938	0.9952	0.9999	1.0001	1.0018
160	0.9880	0.9775	0.9793	0.9790	0.9867	0.9900	0.9905	0.9942
150	0.9918	0.9853	0.9831	0.9844	0.9932	0.9968	0.9969	0.9970
100	0.9913	0.9741	0.9727	0.9740	0.9771	0.9795	0.9794	0.9768
50	0.9971	0.9840	0.9793	0.9827	0.9825	0.9822	0.9822	0.9812
Min	0.9777	0.9741	0.9703	0.9740	0.9771	0.9795	0.9794	0.9768

TABLE C.4: $KPS_{max} : 1 - R^2$ for different PLS components configurations in 1_1 ngram_combination.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9735	0.9694	0.9706	0.9787	0.9839	0.9821	0.9849	0.9849
350	0.9692	0.9583	0.9557	0.9650	0.9660	0.9668	0.9682	0.9682
300	0.9675	0.9621	0.9535	0.9631	0.9655	0.9655	0.9696	0.9681
250	0.9699	0.9554	0.9495	0.9579	0.9571	0.9561	0.9583	0.9615
240	0.9673	0.9555	0.9506	0.9603	0.9580	0.9576	0.9594	0.9592
230	0.9740	0.9640	0.9583	0.9670	0.9688	0.9685	0.9698	0.9698
220	0.9749	0.9653	0.9604	0.9674	0.9678	0.9660	0.9684	0.9712
210	0.9698	0.9609	0.9557	0.9619	0.9621	0.9624	0.9636	0.9646
200	0.9714	0.9606	0.9589	0.9670	0.9648	0.9648	0.9662	0.9671
190	0.9785	0.9688	0.9647	0.9734	0.9742	0.9756	0.9767	0.9785
180	0.9787	0.9695	0.9677	0.9752	0.9746	0.9749	0.9754	0.9769
170	0.9798	0.9634	0.9636	0.9716	0.9718	0.9734	0.9756	0.9752
160	0.9774	0.9687	0.9652	0.9716	0.9720	0.9734	0.9730	0.9721
150	0.9799	0.9633	0.9634	0.9697	0.9682	0.9693	0.9706	0.9700
100	0.9764	0.9619	0.9607	0.9667	0.9666	0.9678	0.9692	0.9696
50	0.9898	0.9742	0.9710	0.9746	0.9765	0.9785	0.9783	0.9784
Min	0.9673	0.9554	0.9495	0.9579	0.9571	0.9561	0.9583	0.9592

TABLE C.5: KPS_{max} : $1 - R^2$ for different PLS components configurations in 1_2 ngram_combination.

Features	**1C**	**2C**	**3C**	**4C**	**5C**	**10C**	**20C**	**50C**
400	0.9650	0.9580	0.9536	0.9639	0.9693	0.9743	0.9786	0.9789
350	0.9695	0.9550	0.9500	0.9591	0.9643	0.9679	0.9763	0.9767
300	0.9653	0.9602	0.9544	0.9635	0.9685	0.9701	0.9778	0.9787
250	0.9692	0.9584	0.9549	0.9633	0.9630	0.9635	0.9669	0.9705
240	0.9658	0.9587	0.9542	0.9633	0.9626	0.9646	0.9666	0.9664
230	0.9701	0.9581	0.9535	0.9627	0.9655	0.9724	0.9758	0.9763
220	0.9743	0.9651	0.9627	0.9725	0.9739	0.9771	0.9826	0.9861
210	0.9709	0.9617	0.9590	0.9675	0.9719	0.9721	0.9775	0.9796
200	0.9750	0.9609	0.9581	0.9654	0.9664	0.9699	0.9724	0.9735
190	0.9803	0.9722	0.9686	0.9789	0.9818	0.9845	0.9867	0.9851
180	0.9803	0.9702	0.9690	0.9766	0.9781	0.9802	0.9841	0.9846
170	0.9759	0.9628	0.9611	0.9691	0.9696	0.9728	0.9748	0.9765
160	0.9771	0.9668	0.9652	0.9728	0.9712	0.9733	0.9734	0.9739
150	0.9810	0.9771	0.9688	0.9753	0.9773	0.9774	0.9792	0.9803
100	0.9797	0.9670	0.9641	0.9695	0.9707	0.9772	0.9790	0.9773
50	0.9899	0.9695	0.9680	0.9742	0.9772	0.9862	0.9866	0.9894
Min	0.9650	0.9550	0.9500	0.9591	0.9626	0.9635	0.9666	0.9664

TABLE C.6: KPS_{max} : $1 - R^2$ for different PLS components configurations in 1_3 ngram_combination.

C.2 Multilayer Perceptron Results

C.2.1 Identity Activation Function

Features	(2,100)	(5,100)	(7,100)	(10,100)	(15,100)	(25,100)
500	0.9140	0.9132	0.9191	0.9163	0.9676	0.9162
450	0.9172	0.9384	0.9170	0.9124	0.9181	0.9136
400	0.9126	0.9132	0.9118	0.9157	0.9316	0.9153
350	0.9164	0.9196	0.9155	0.9189	0.9158	0.9139
300	0.9269	0.9163	0.9195	0.9189	0.9213	0.9216
250	0.9197	0.9309	0.9278	0.9599	0.9170	0.9180
240	0.9210	0.9372	0.9230	0.9070	0.9161	0.9208
230	0.9237	0.9192	0.9590	0.9221	0.9237	0.9224
220	0.9180	0.9221	0.9171	0.8973	0.9192	0.9211
210	0.9331	0.9215	0.9227	0.9183	0.9174	0.9220
200	0.9236	0.9365	0.9247	0.9025	0.9254	0.9111
190	0.9178	0.9169	0.9222	0.9385	0.9090	0.9215
180	0.9212	0.9234	0.9225	0.9833	0.9215	0.9234
170	0.9313	0.9332	0.9645	0.9160	0.9426	0.9267
160	0.9907	0.9317	0.9279	0.9280	0.9268	0.9309
150	0.9313	0.9329	0.9349	0.9344	0.9370	0.9362
100	0.9471	0.9369	0.9346	0.9427	0.9438	0.9423
50	0.9568	0.9727	0.9576	0.9582	0.9973	0.9579
Min	0.9126	0.9132	0.9118	0.8973	0.9090	0.9111

TABLE C.7: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	**(2,100)**	**(5,100)**	**(7,100)**	**(10,100)**	**(15,100)**	**(25,100)**
500	0.9051	0.9022	0.9006	0.9056	0.9471	0.9009
450	0.9100	0.9249	0.9017	0.8965	0.9044	0.9041
400	0.9110	0.9023	0.9129	0.9129	0.9228	0.9103
350	0.9075	0.9140	0.9127	0.9054	0.9015	0.9080
300	0.9131	0.9198	0.9014	0.9162	0.9094	0.9114
250	0.9095	0.9575	0.9232	0.9276	0.9136	0.9147
240	0.9069	0.9930	0.9097	0.9164	0.9043	0.9072
230	0.9093	0.9181	0.9813	0.9020	0.9105	0.9111
220	0.9126	0.9152	0.9107	0.9059	0.9180	0.9164
210	0.9165	0.9144	0.9162	0.9140	0.9148	0.9179
200	0.9197	0.9318	0.9235	0.8985	0.9203	0.9070
190	0.9213	0.9306	0.9269	0.9286	0.9114	0.9215
180	0.9184	0.9552	0.9224	0.9351	0.9182	0.9195
170	0.9218	0.9240	0.9169	0.9098	0.9242	0.9216
160	0.9493	0.9217	0.9220	0.9395	0.9226	0.9245
150	0.9282	0.9286	0.9316	0.9304	0.9323	0.9331
100	0.9459	0.9338	0.9289	0.9358	0.9365	0.9363
50	0.9590	0.9551	0.9577	0.9583	0.9632	0.9577
Min	0.9051	0.9022	0.9006	0.8965	0.9015	0.9009

TABLE C.8: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,100)**	**(5,100)**	**(7,100)**	**(10,100)**	**(15,100)**	**(25,100)**
500	0.9084	0.9111	0.9085	0.9068	0.9652	0.9074
450	0.9069	0.9271	0.9040	0.9067	0.9068	0.9083
400	0.9056	0.9073	0.9113	0.9129	0.9177	0.9103
350	0.9086	0.9122	0.9128	0.9133	0.9095	0.9100
300	0.9129	0.9168	0.9107	0.9127	0.9127	0.9109
250	0.9078	0.9179	0.9171	0.9197	0.9096	0.9127
240	0.9056	0.9363	0.9112	0.8903	0.9048	0.9055
230	0.9112	0.9130	0.9874	0.9078	0.9124	0.9131
220	0.9159	0.9162	0.8965	0.9072	0.9143	0.9135
210	0.8985	0.9142	0.9139	0.9122	0.9076	0.9154
200	0.9115	0.9260	0.9166	0.8962	0.9174	0.9024
190	0.9148	0.9092	0.9381	0.9233	0.8931	0.9165
180	0.9175	0.9501	0.9307	0.9354	0.9164	0.9183
170	0.9198	0.9233	0.9259	0.9275	0.9383	0.9210
160	0.9905	0.9228	0.9200	0.9177	0.9141	0.9203
150	0.9260	0.9222	0.9267	0.9251	0.9267	0.9259
100	0.9429	0.9350	0.9334	0.9400	0.9416	0.9421
50	0.9567	0.9583	0.9573	0.9600	0.9621	0.9582
Min	0.8985	0.9073	0.8965	0.8903	0.8931	0.9024

TABLE C.9: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_ombination.

Features	(2,50)	(5,50)	(7,25)	(10,100)	(15,25)	(25,25)
500	1.0267	1.0347	1.0017	1.0795	1.0446	1.0220
450	1.0039	1.0508	1.0034	1.0497	1.0127	1.0346
400	1.0346	1.0359	1.0119	1.0608	1.0151	1.0056
350	1.0246	1.0556	1.0557	0.9999	0.9922	1.0397
300	1.0245	1.0585	1.0506	1.0920	1.0377	1.1117
250	1.0227	1.0301	1.0049	0.9891	1.0071	1.0115
240	1.0185	1.0159	1.0091	1.0083	1.0216	1.0005
230	1.0287	1.0272	1.0212	1.0318	0.9904	1.1888
220	1.0435	1.0358	1.0296	1.0235	0.9823	0.9885
210	1.0020	1.0137	1.0269	1.0571	1.0231	1.0159
200	1.0197	1.0046	1.0125	1.0095	0.9968	1.0144
190	1.0452	1.0277	0.9895	1.0271	1.0077	1.0437
180	1.0391	1.0267	1.0356	1.0124	1.0002	0.9883
170	1.0301	0.9958	1.0126	1.0783	1.0029	1.0167
160	1.0148	1.0312	1.0166	1.0278	1.0112	1.0214
150	1.0139	1.0328	0.9943	1.0427	1.0155	1.0338
100	1.0255	1.0230	1.0224	1.0254	1.0108	1.0035
50	1.0150	1.0024	1.0039	1.0173	1.0195	1.0028
Min	1.0020	0.9958	0.9895	0.9891	0.9823	0.9883

TABLE C.10: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	(2,50)	(5,50)	(7,25)	(10,100)	(15,25)	(25,25)
500	1.0073	1.0309	0.9955	1.0594	1.0188	1.0333
450	1.0551	1.0543	1.0025	1.0529	1.0144	1.0644
400	0.9920	1.0260	1.0006	1.0390	1.0110	1.0016
350	1.0126	1.0525	1.0000	0.9994	0.9817	0.9894
300	1.0174	1.0176	1.0182	1.0495	1.0001	1.1041
250	1.0134	0.9884	0.9884	0.9935	1.0033	1.0079
240	1.0086	1.0033	1.0126	0.9668	1.0189	1.0058
230	1.0256	0.9919	1.0085	1.0118	1.0065	1.1429
220	1.0273	1.0146	1.0127	1.0075	1.0289	1.0231
210	1.0144	1.0096	1.0245	1.0173	1.0383	1.0042
200	1.0460	0.9896	0.9997	1.0292	1.0048	1.0251
190	1.0551	1.0043	0.9785	1.0251	1.0174	0.9865
180	1.0262	0.9746	0.9999	1.0095	1.0223	0.9879
170	1.0240	0.9981	1.0112	1.0101	1.0007	0.9920
160	1.0112	1.0278	1.0205	1.0175	1.0244	1.0122
150	1.0137	1.0250	1.0235	1.0200	0.9966	1.0770
100	0.9992	1.0217	1.0185	1.0163	1.0655	0.9942
50	1.0107	0.9981	0.9856	1.0095	1.0119	1.0088
Min	0.9920	0.9746	0.9785	0.9668	0.9817	0.9865

TABLE C.11: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,50)**	**(5,50)**	**(7,25)**	**(10,100)**	**(15,25)**	**(25,25)**
500	0.9794	1.0268	0.9988	1.0737	1.0264	1.0214
450	1.0199	1.0743	1.0100	1.0431	1.0138	1.0570
400	1.0189	1.0080	1.0114	1.0391	1.0162	1.0073
350	1.0180	1.0444	0.9970	1.0170	0.9850	1.0103
300	1.0250	1.0246	1.0164	1.0580	1.0287	1.0573
250	0.9995	1.0210	0.9854	1.0184	0.9970	1.0190
240	1.0151	1.0049	1.0216	0.9734	1.0128	1.0072
230	1.0018	1.0106	1.0147	0.9997	1.0020	1.0900
220	1.0287	1.0258	1.0411	0.9917	1.0117	1.0062
210	0.9994	1.0056	1.0282	1.0308	1.0138	1.0078
200	1.0299	1.0073	1.0038	1.0305	1.0006	1.0364
190	1.0308	1.0097	1.0002	1.0001	0.9910	1.0012
180	1.0297	0.9849	0.9890	1.0170	1.0309	0.9807
170	1.0117	1.0122	1.0176	1.0071	1.0017	1.0150
160	1.0101	1.0136	1.0353	1.0367	1.0167	1.0261
150	1.0219	1.0261	1.0149	1.0372	1.0036	1.0391
100	1.0180	1.0177	1.0581	1.0174	0.9940	1.0041
50	1.0116	0.9992	1.0121	1.0201	1.0102	1.0072
Min	0.9794	0.9849	0.9854	0.9734	0.9850	0.9807

TABLE C.12: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

C.2.2 Logistic Activation Function

Features	(2,75)	(5,75)	(7,75)	(10,75)	(15,75)	(25,75)
500	0.9621	1.0000	1.0000	0.9713	1.0000	1.0000
450	0.9709	0.9567	1.0000	1.0000	1.0006	1.0000
400	0.9267	1.0000	1.0000	1.0000	1.0000	1.0000
350	1.0000	0.9297	0.9422	1.0000	1.0000	1.0000
300	0.9241	1.0000	1.0000	1.0000	1.0000	0.9450
250	1.0000	1.0000	1.0000	0.9190	1.0000	1.0000
240	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
230	0.9315	1.0000	1.0000	0.9359	1.0000	1.0000
220	0.9147	1.0000	0.9123	0.9382	1.0000	1.0000
210	0.9386	1.0000	1.0000	0.9268	1.0000	1.0000
200	0.9221	1.0000	1.0000	1.0000	0.9319	0.9359
190	1.0000	1.0000	0.9283	1.0000	1.0000	0.9363
180	0.9221	1.0000	0.9098	0.9163	1.0000	1.0000
170	1.0000	1.0000	0.9193	1.0000	1.0000	1.0000
160	0.9339	0.9333	1.0000	0.9267	1.0000	1.0000
150	1.0000	0.9428	0.9253	1.0000	1.0000	1.0000
100	0.9522	0.9286	0.9323	0.9211	1.0000	1.0000
50	1.0000	1.0000	0.9421	1.0000	1.0000	1.0000
Min	0.9147	0.9286	0.9098	0.9163	0.9319	0.9359

TABLE C.13: $KPS_{min} : 1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	**(2,75)**	**(5,75)**	**(7,75)**	**(10,75)**	**(15,75)**	**(25,75)**
500	0.9665	1.0000	1.0000	0.9495	1.0000	1.0000
450	0.9469	0.9004	1.0000	1.0000	0.9784	1.0000
400	0.9168	1.0000	1.0000	1.0000	1.0000	1.0000
350	1.0000	0.9280	0.9154	1.0000	1.0000	1.0000
300	0.9294	1.0000	1.0000	1.0000	1.0000	0.9210
250	1.0000	1.0000	1.0000	0.9065	1.0000	1.0000
240	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
230	0.9097	1.0000	1.0000	0.9086	1.0000	1.0000
220	0.9080	1.0000	0.9145	0.9076	1.0000	1.0000
210	0.9121	1.0000	1.0000	0.9165	1.0000	1.0000
200	0.9112	1.0000	1.0000	1.0000	0.9288	0.9358
190	1.0000	1.0000	0.9024	1.0000	1.0000	0.9267
180	0.9118	1.0000	0.9008	0.9176	1.0000	1.0000
170	1.0000	1.0000	0.9126	1.0000	1.0000	1.0000
160	1.0000	0.9090	1.0000	0.9111	1.0000	1.0000
150	1.0000	0.9185	0.9180	1.0000	1.0000	1.0000
100	0.9344	0.9154	0.9103	0.9142	1.0000	1.0000
50	1.0000	1.0000	0.9414	1.0000	1.0000	1.0000
Min	0.9080	0.9004	0.9008	0.9065	0.9288	0.9210

TABLE C.14: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,75)**	**(5,75)**	**(7,75)**	**(10,75)**	**(15,75)**	**(25,75)**
500	0.9618	1.0000	1.0000	0.9466	1.0000	1.0000
450	0.9719	0.9097	1.0000	1.0000	0.9697	1.0000
400	0.9419	1.0000	1.0000	1.0000	1.0000	1.0000
350	1.0000	0.9248	0.9089	1.0000	1.0000	1.0000
300	0.9458	1.0000	1.0000	1.0000	1.0000	0.9458
250	1.0000	1.0000	1.0000	0.9170	1.0000	1.0000
240	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
230	0.9146	1.0000	1.0000	0.9153	1.0000	1.0000
220	0.9217	1.0000	0.9108	0.9040	1.0000	1.0000
210	0.9069	1.0000	1.0000	0.9127	1.0000	1.0000
200	0.9118	1.0000	1.0000	1.0000	0.9179	0.9052
190	1.0000	1.0000	0.8965	1.0000	1.0000	0.9257
180	0.8951	1.0000	0.9001	0.9075	1.0000	1.0000
170	1.0000	1.0000	0.9239	1.0000	1.0000	1.0000
160	1.0000	0.9143	1.0000	0.9249	1.0000	1.0000
150	1.0000	0.9130	0.9147	1.0000	1.0000	1.0000
100	0.9362	0.9236	0.9239	0.9220	1.0000	1.0000
50	1.0000	1.0000	0.9532	1.0000	1.0000	1.0000
Min	0.8951	0.9097	0.8965	0.9040	0.9179	0.9052

TABLE C.15: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

Features	(2,100)	(5,50)	(7,50)	(10,50)	(15,25)	(25,25)
500	1.0743	0.9999	0.9999	0.9998	0.9999	1.0868
450	1.0523	1.0542	0.9999	0.9999	1.0557	0.9999
400	1.0601	0.9999	0.9999	0.9999	0.9999	0.9999
350	0.9999	1.0725	1.0597	0.9999	0.9999	0.9999
300	1.0458	0.9999	0.9999	0.9999	0.9999	1.0005
250	0.9999	0.9999	0.9999	0.9860	0.9999	0.9999
240	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
230	0.9990	0.9999	0.9999	1.0106	0.9999	0.9999
220	1.0549	0.9999	1.0138	1.0105	0.9999	0.9999
210	1.0366	0.9999	0.9999	1.0306	0.9999	0.9999
200	1.0462	0.9999	0.9999	0.9999	1.0502	1.0118
190	0.9999	0.9999	0.9895	0.9999	0.9999	1.0472
180	1.0109	0.9999	1.0751	1.0376	0.9999	0.9999
170	0.9999	0.9999	1.0449	0.9999	0.9999	0.9999
160	1.0366	1.0389	0.9999	1.0643	0.9999	0.9999
150	0.9999	1.0399	1.0185	0.9999	0.9999	0.9999
100	1.0420	1.0124	1.0353	1.0052	0.9999	0.9999
50	0.9999	0.9999	1.0054	0.9999	0.9999	0.9999
Min	0.9990	0.9999	0.9895	0.9860	0.9999	0.9999

TABLE C.16: $KPS_{max} : 1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	(2,100)	(5,50)	(7,50)	(10,50)	(15,25)	(25,25)
500	1.0699	0.9999	0.9999	1.0427	0.9999	0.9999
450	1.0213	0.9961	0.9999	0.9999	1.0458	0.9999
400	1.0155	0.9999	0.9999	0.9999	0.9999	0.9999
350	0.9999	1.0375	1.0198	0.9999	0.9999	0.9999
300	1.0317	0.9999	0.9999	0.9999	0.9999	0.9939
250	0.9999	0.9999	0.9999	0.9800	0.9999	0.9999
240	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
230	0.9916	0.9999	0.9999	1.0661	0.9999	0.9999
220	1.0363	0.9999	0.9689	1.0264	0.9999	0.9999
210	1.0037	0.9999	0.9999	0.9985	0.9999	0.9999
200	1.0143	0.9999	0.9999	0.9999	1.0307	1.0374
190	0.9999	0.9999	1.0195	0.9999	0.9999	0.9842
180	0.9922	0.9999	1.0315	1.0083	0.9999	0.9999
170	0.9999	0.9999	1.0521	0.9999	0.9999	0.9999
160	0.9999	1.0238	0.9999	1.0560	0.9999	0.9999
150	0.9999	1.0431	0.9895	0.9999	0.9999	0.9999
100	1.0557	0.9852	1.0226	0.9916	0.9999	0.9999
50	0.9999	0.9999	1.0191	0.9999	0.9999	0.9999
Min	0.9916	0.9852	0.9689	0.9800	0.9999	0.9842

TABLE C.17: $KPS_{max} : 1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,100)**	**(5,50)**	**(7,50)**	**(10,50)**	**(15,25)**	**(25,25)**
500	1.1069	0.9999	0.9999	0.9971	0.9999	0.9999
450	1.0484	0.9936	0.9999	0.9999	1.0558	0.9999
400	1.0381	0.9999	0.9999	0.9999	0.9999	0.9999
350	0.9999	1.0324	1.0110	0.9999	0.9999	0.9999
300	1.0419	0.9999	0.9999	0.9999	0.9999	0.9944
250	0.9999	0.9999	0.9999	1.0346	0.9999	0.9999
240	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
230	1.0031	0.9999	0.9999	1.0201	0.9999	0.9999
220	1.0586	0.9999	0.9834	1.0299	0.9999	0.9999
210	1.0215	0.9999	0.9999	1.0203	0.9999	0.9999
200	1.0034	0.9999	0.9999	0.9999	1.0253	1.0068
190	0.9999	0.9999	1.0264	0.9999	0.9999	1.0391
180	0.9945	0.9999	1.0356	0.9945	0.9999	0.9999
170	0.9999	0.9999	1.0728	0.9999	0.9999	0.9999
160	0.9999	1.0327	0.9999	1.0450	0.9999	0.9999
150	0.9999	1.0744	0.9948	0.9999	0.9999	0.9999
100	1.0263	1.0002	1.0554	1.0088	0.9999	0.9999
50	0.9999	0.9999	1.0033	0.9999	0.9999	0.9999
Min	0.9945	0.9936	0.9834	0.9945	0.9999	0.9944

TABLE C.18: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

C.2.3 Hyperbolic Tangent Activation Function

Features	(2,50)	(5,50)	(7,75)	(10,50)	(15,50)	(25,10)
500	0.9406	1.0000	1.0000	1.0000	1.0000	1.0000
450	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
400	1.0000	1.0000	1.0000	0.9316	1.0000	1.0000
350	1.0000	0.9559	1.0000	1.0000	1.0000	1.0000
300	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
250	0.9442	1.0000	1.0000	1.0000	1.0000	1.0000
240	1.0000	1.0000	1.0000	0.9698	1.0000	1.0000
230	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
220	0.9333	1.0000	1.0000	0.9556	1.0000	1.0000
210	1.0000	1.0000	1.0000	1.0000	0.9298	1.0000
200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
190	0.9388	1.0000	1.0000	1.0000	1.0000	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
160	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	1.0000	1.0000	1.0000	1.0000	0.9937	1.0000
100	1.0000	1.0000	1.0000	0.9582	1.0000	1.0000
50	1.0000	1.0000	1.0000	0.9761	1.0000	1.0000
Min	0.9333	0.9559	1.0000	0.9316	0.9298	1.0000

TABLE C.19: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	**(2,50)**	**(5,50)**	**(7,75)**	**(10,50)**	**(15,50)**	**(25,10)**
500	0.9306	1.0000	1.0000	1.0000	1.0000	1.0000
450	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
400	1.0000	1.0000	1.0000	0.9477	1.0000	1.0000
350	1.0000	0.9344	1.0000	1.0000	1.0000	1.0000
300	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
250	0.9388	1.0000	1.0000	1.0000	1.0000	1.0000
240	1.0000	1.0000	1.0000	0.9610	1.0000	1.0000
230	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
220	0.9256	1.0000	1.0000	0.9235	1.0000	1.0000
210	1.0000	1.0000	0.9754	1.0000	0.9226	1.0000
200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
190	0.9396	1.0000	1.0000	1.0000	1.0000	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
160	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	1.0000	1.0000	1.0000	0.9486	1.0000	1.0000
50	1.0000	1.0000	1.0000	0.9956	1.0000	1.0000
Min	0.9256	0.9344	0.9754	0.9235	0.9226	1.0000

TABLE C.20: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,50)**	**(5,50)**	**(7,75)**	**(10,50)**	**(15,50)**	**(25,10)**
500	0.9266	1.0000	1.0000	1.0000	1.0000	1.0000
450	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
400	1.0000	1.0000	1.0000	0.9335	1.0000	1.0000
350	1.0000	0.9324	1.0000	1.0000	1.0000	1.0000
300	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
250	0.9517	1.0000	1.0000	1.0000	1.0000	1.0000
240	1.0000	1.0000	1.0000	0.9268	1.0000	1.0000
230	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
220	0.9303	1.0000	1.0000	0.9264	1.0000	1.0000
210	1.0000	1.0000	0.9786	1.0000	0.9259	1.0000
200	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
190	0.9368	1.0000	1.0000	1.0000	1.0000	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
160	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
150	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
100	1.0000	1.0000	0.9726	0.9527	1.0000	1.0000
50	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Min	0.9266	0.9324	0.9726	0.9264	0.9259	1.0000

TABLE C.21: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

Features	(2,50)	(5,10)	(7,50)	(10,10)	(15,10)	(25,10)
500	1.0355	0.9999	0.9999	0.9999	0.9999	0.9999
450	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
400	0.9999	0.9999	0.9999	1.0475	0.9999	0.9999
350	0.9999	0.9981	0.9999	0.9999	0.9999	0.9999
300	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
250	1.0544	0.9999	0.9999	0.9999	0.9999	0.9999
240	0.9999	0.9999	0.9999	1.0034	0.9999	0.9999
230	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
220	1.0582	0.9999	0.9999	1.0163	0.9999	0.9999
210	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
200	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
190	1.0729	0.9999	0.9999	0.9999	0.9999	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
160	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
150	0.9999	0.9999	0.9999	0.9999	1.0914	0.9999
100	0.9999	0.9999	0.9999	1.0494	0.9999	0.9999
50	0.9999	0.9999	0.9999	0.9845	0.9999	0.9999
Min	0.9999	0.9981	0.9999	0.9845	0.9999	0.9999

TABLE C.22: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	(2,50)	(5,10)	(7,50)	(10,10)	(15,10)	(25,10)
500	1.0342	0.9999	0.9999	0.9999	0.9999	0.9999
450	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
400	0.9999	0.9999	0.9999	1.0279	0.9999	0.9999
350	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
300	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
250	1.0330	0.9999	0.9999	0.9999	0.9999	0.9999
240	0.9999	0.9999	0.9999	1.0193	0.9999	0.9999
230	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
220	1.0321	0.9999	0.9999	1.0392	0.9999	0.9999
210	0.9999	0.9999	0.9965	0.9999	0.9945	0.9999
200	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
190	1.0847	0.9999	0.9999	0.9999	0.9999	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
160	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
150	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
100	0.9999	0.9999	0.9999	1.0199	0.9999	0.9999
50	0.9999	0.9999	0.9999	1.0011	0.9999	0.9999
Min	0.9999	0.9999	0.9965	0.9999	0.9945	0.9999

TABLE C.23: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	(2,50)	(5,10)	(7,50)	(10,10)	(15,10)	(25,10)
500	1.0303	0.9999	0.9999	0.9999	0.9999	0.9999
450	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
400	0.9999	0.9999	0.9999	1.0280	0.9999	0.9999
350	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
300	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
250	0.9879	0.9999	0.9999	0.9999	0.9999	0.9999
240	0.9999	0.9999	0.9999	1.0038	0.9999	0.9999
230	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
220	1.0215	0.9999	0.9999	1.0322	0.9999	0.9999
210	0.9999	0.9999	1.0068	0.9999	0.9999	0.9999
200	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
190	1.0846	0.9999	0.9999	0.9999	0.9999	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
160	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
150	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
100	0.9999	0.9999	1.0108	1.0138	0.9999	0.9999
50	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
Min	0.9879	0.9999	0.9999	0.9999	0.9999	0.9999

TABLE C.24: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

C.2.4 Rectified Linear Unit Activation Function

Features	(2,100)	(5,100)	(7,100)	(10,100)	(15,100)	(25,100)
500	1.0000	0.9179	1.0000	1.0000	0.9419	1.0000
450	1.0000	0.9604	1.0000	1.0000	1.0000	0.9128
400	1.0000	0.9123	0.9176	0.9183	1.0000	0.9172
350	0.9140	0.9200	1.0000	0.9994	1.0000	1.0000
300	0.9203	1.0000	1.0000	1.0000	0.9188	0.9352
250	1.0000	1.0000	0.9202	1.0000	0.9153	0.9112
240	0.9256	1.0000	1.0000	1.0000	0.9143	0.9180
230	1.0000	0.9192	1.0000	0.9147	1.0000	0.9172
220	0.9198	0.9205	1.0000	1.0007	0.9855	1.0000
210	1.0000	1.0000	0.9209	0.9259	0.9292	0.9197
200	1.0000	0.9892	0.9209	1.0000	1.0000	0.9209
190	1.0000	1.0000	1.0000	1.0000	0.9134	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	0.9303	1.0000	1.0000	0.9417	1.0000	0.9950
160	1.0000	0.9300	1.0000	1.0000	1.0000	0.9127
150	0.9275	1.0000	1.0000	0.9499	0.9353	1.0000
100	0.9448	1.0000	0.9435	0.9342	0.9435	0.9438
50	1.0000	0.9476	1.0000	1.0000	1.0000	0.9561
Min	0.9140	0.9123	0.9176	0.9147	0.9134	0.9112

TABLE C.25: $KPS_{min} : 1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	**(2,100)**	**(5,100)**	**(7,100)**	**(10,100)**	**(15,100)**	**(25,100)**
500	0.9085	0.9071	1.0000	1.0000	0.9134	1.0000
450	1.0000	0.9566	1.0000	1.0000	1.0000	0.9048
400	1.0000	0.8944	0.9148	0.9142	1.0000	0.9104
350	0.9056	0.9165	1.0000	1.0000	1.0000	1.0000
300	0.9187	1.0000	1.0000	1.0000	0.9081	0.9443
250	1.0000	1.0000	0.9073	1.0000	0.9145	0.9107
240	0.9091	1.0000	1.0000	1.0000	0.9019	0.9078
230	1.0000	0.9109	1.0000	0.9122	1.0000	0.9097
220	0.9138	0.9124	1.0000	1.0032	1.0000	1.0000
210	1.0000	1.0000	0.9136	0.9132	0.9124	0.9008
200	1.0000	1.0000	0.9065	1.0000	1.0000	0.9148
190	1.0000	1.0000	1.0000	1.0000	0.9121	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	0.9202	1.0000	1.0000	0.9256	1.0000	1.0000
160	1.0000	0.9170	1.0000	1.0000	1.0000	0.9071
150	0.9224	1.0000	1.0000	0.9824	0.9297	1.0000
100	0.9382	1.0000	0.9349	0.9172	0.9364	0.9354
50	1.0000	0.9575	1.0000	1.0000	1.0000	0.9557
Min	0.9056	0.8944	0.9065	0.9122	0.9019	0.9008

TABLE C.26: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,100)**	**(5,100)**	**(7,100)**	**(10,100)**	**(15,100)**	**(25,100)**
500	0.9314	0.9112	1.0000	1.0000	0.9422	1.0000
450	1.0000	0.9451	1.0000	1.0000	1.0000	0.9077
400	1.0000	0.9127	0.9187	0.9165	1.0000	0.9096
350	0.9096	0.9099	1.0000	1.0057	1.0000	1.0000
300	0.9208	1.0000	1.0000	1.0000	0.9076	0.9549
250	1.0000	1.0000	0.9013	1.0000	0.9068	0.9035
240	0.9134	1.0000	1.0000	1.0000	0.8976	0.9029
230	1.0000	0.9083	1.0000	0.9010	1.0000	0.9085
220	0.9150	0.9145	1.0000	1.0000	1.0000	1.0000
210	1.0000	1.0000	0.9135	0.9109	0.9110	0.9337
200	1.0000	1.0000	0.9124	1.0000	1.0000	0.9149
190	1.0000	1.0000	1.0000	1.0000	0.8934	1.0000
180	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
170	0.9234	1.0000	1.0000	0.9325	1.0000	1.0000
160	1.0000	0.9198	1.0000	1.0000	1.0000	0.9084
150	0.9219	1.0000	1.0000	0.9236	0.9247	1.0000
100	0.9401	1.0000	0.9544	0.9246	0.9366	0.9412
50	1.0000	0.9514	1.0000	1.0000	1.0000	0.9595
Min	0.9096	0.9083	0.9013	0.9010	0.8934	0.9029

TABLE C.27: KPS_{min} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

Features	(2,25)	(5,25)	(7,25)	(10,25)	(15,25)	(25,100)
500	0.9999	0.9844	0.9999	0.9999	1.0187	0.9999
450	0.9999	1.0336	0.9999	0.9999	0.9999	1.1220
400	0.9999	1.0193	1.0294	0.9892	0.9999	1.0655
350	1.0050	1.0753	0.9999	1.0010	0.9999	0.9999
300	1.0510	0.9999	0.9999	0.9999	1.0138	1.0164
250	0.9999	0.9999	1.0319	0.9999	1.0477	1.0460
240	0.9948	0.9999	0.9999	0.9999	1.0030	1.0515
230	0.9999	0.9955	0.9999	1.0211	0.9999	1.0724
220	0.9996	1.0171	0.9999	1.0060	1.0138	0.9999
210	0.9999	0.9999	1.0187	1.0101	1.0173	1.0275
200	0.9999	1.0013	1.0253	0.9999	0.9999	1.0960
190	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	0.9910	0.9999	0.9999	0.9966	0.9999	1.0018
160	0.9999	1.0101	0.9999	0.9999	0.9999	1.0148
150	1.0263	0.9999	0.9999	0.9907	1.0109	0.9999
100	0.9975	0.9999	1.0222	0.9998	0.9869	1.0245
50	0.9999	1.0167	0.9999	0.9999	0.9999	1.0221
Min	0.9910	0.9844	0.9999	0.9892	0.9869	0.9999

TABLE C.28: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_1 ngram_combination.

Features	(2,25)	(5,25)	(7,25)	(10,25)	(15,25)	(25,100)
500	1.0088	1.0375	0.9999	0.9999	1.0125	0.9999
450	0.9999	0.9899	0.9999	0.9999	0.9999	1.1036
400	0.9999	1.0036	1.0046	0.9770	0.9999	1.0538
350	1.0056	1.0373	0.9999	0.9999	0.9999	0.9999
300	1.0168	0.9999	0.9999	0.9999	1.0247	0.9811
250	0.9999	0.9999	1.0313	0.9999	1.0655	0.9710
240	0.9906	0.9999	0.9999	0.9999	1.0248	1.0207
230	0.9999	0.9892	0.9999	1.0407	0.9999	1.0308
220	1.0035	1.0027	0.9999	1.0069	0.9999	0.9999
210	0.9999	0.9999	1.0281	1.0144	1.0226	1.0035
200	0.9999	0.9999	1.0117	0.9999	0.9999	1.0605
190	0.9999	0.9999	0.9999	0.9999	1.0096	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	1.0078	0.9999	0.9999	1.0100	0.9999	0.9999
160	0.9999	1.0111	0.9999	0.9999	0.9999	1.0138
150	1.0167	0.9999	0.9999	1.0265	1.0121	0.9999
100	1.0116	0.9999	1.0056	1.0103	0.9855	1.0157
50	0.9999	1.0202	0.9999	0.9999	0.9999	1.0121
Min	0.9906	0.9892	0.9999	0.9770	0.9855	0.9710

TABLE C.29: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_2 ngram_combination.

Features	**(2,25)**	**(5,25)**	**(7,25)**	**(10,25)**	**(15,25)**	**(25,100)**
500	1.0052	1.0129	0.9999	0.9999	1.0211	0.9999
450	0.9999	0.9868	0.9999	0.9999	0.9999	1.0522
400	0.9999	1.0170	1.0341	1.0017	0.9999	1.0469
350	1.0082	1.0115	0.9999	1.0020	0.9999	0.9999
300	1.0383	0.9999	0.9999	0.9999	1.0182	0.9993
250	0.9999	0.9999	1.0157	0.9999	1.0500	1.0226
240	0.9963	0.9999	0.9999	0.9999	1.0029	1.0208
230	0.9999	0.9864	0.9999	1.0404	0.9999	1.0462
220	1.0146	0.9973	0.9999	0.9999	0.9999	0.9999
210	0.9999	0.9999	1.0271	1.0063	1.0139	0.9979
200	0.9999	0.9999	0.9654	0.9999	0.9999	1.0603
190	0.9999	0.9999	0.9999	0.9999	0.9965	0.9999
180	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
170	0.9948	0.9999	0.9999	1.0172	0.9999	0.9999
160	0.9999	0.9986	0.9999	0.9999	0.9999	1.0245
150	0.9974	0.9999	0.9999	0.9932	1.0201	0.9999
100	1.0207	0.9999	1.0156	1.0078	0.9915	1.0198
50	0.9999	1.0171	0.9999	0.9999	0.9999	1.0121
Min	0.9948	0.9864	0.9654	0.9932	0.9915	0.9979

TABLE C.30: KPS_{max} : $1 - R^2$ for the best MLP (neurons, epochs) and $\alpha = 1x10^{-3}$ configurations in 1_3 ngram_combination.

Appendix D

Problematic Cases in Clinical Trials

D.1 Response Variables in CT Problematic Cases

In this Appendix we are going to show the KPS *min*, and *max* response variables related to the problematic cases in clinical trials.

- *NCT*00001880: $[0, 40]$
- *NCT*00001880: $[0, 40]$
- *NCT*00002628: $[0, 20]$
- *NCT*00010010: $[90, 100]$
- *NCT*00034827: $[10, 40]$
- *NCT*00079625: $[10, 20]$
- *NCT*00130507: $[0, 60]$
- *NCT*00131963: $[0, 40]$
- *NCT*00135018: $[0, 0]$
- *NCT*00162929: $[0, 40]$
- *NCT*00195013: $[100, 100]$
- *NCT*00197522: $[0, 40]$
- *NCT*00239343: $[30, 60]$
- *NCT*00263705: $[30, 40]$
- *NCT*00264082: $[10, 40]$
- *NCT*00295620: $[0, 40]$
- *NCT*00307229: $[0, 40]$
- *NCT*00309530: $[0, 40]$
- *NCT*00309543: $[0, 40]$
- *NCT*00309933: $[10, 20]$
- *NCT*00348699: $[10, 40]$
- *NCT*00412022: $[0, 40]$
- *NCT*00429507: $[30, 40]$
- *NCT*00509691: $[0, 0]$
- *NCT*00516269: $[10, 20]$
- *NCT*00539968: $[50, 60]$
- *NCT*00578006: $[0, 40]$
- *NCT*00637325: $[0, 40]$
- *NCT*00707707: $[0, 40]$
- *NCT*00713141: $[30, 40]$
- *NCT*00759642: $[50, 60]$
- *NCT*00774878: $[0, 40]$
- *NCT*00811369: $[0, 40]$
- *NCT*00824733: $[0, 0]$
- *NCT*00825734: $[50, 60]$
- *NCT*00826085: $[0, 40]$
- *NCT*00841828: $[0, 0]$
- *NCT*00863122: $[60, 60]$
- *NCT*00896727: $[30, 40]$
- *NCT*00923936: $[10, 20]$
- *NCT*00942331: $[0, 0]$
- *NCT*00949013: $[30, 40]$
- *NCT*00951574: $[0, 40]$

- *NCT*00967577: [0, 40]
- *NCT*00968968: [0, 40]
- *NCT*00997529: [0, 0]
- *NCT*01077453: [0, 0]
- *NCT*01091883: [10, 10]
- *NCT*01105650: [50, 50]
- *NCT*01132560: [0, 0]
- *NCT*01159067: [0, 40]
- *NCT*01190345: [0, 0]
- *NCT*01198158: [0, 0]
- *NCT*01306045: [30, 40]
- *NCT*01315119: [0, 0]
- *NCT*01320592: [0, 20]
- *NCT*01320787: [0, 0]
- *NCT*01351909: [0, 0]
- *NCT*01384253: [0, 20]
- *NCT*01396655: [10, 20]
- *NCT*01432002: [30, 40]
- *NCT*01483001: [100, 100]
- *NCT*01495663: [0, 20]
- *NCT*01503372: [50, 60]
- *NCT*01513356: [50, 60]
- *NCT*01530373: [0, 40]
- *NCT*01534455: [0, 60]
- *NCT*01596647: [0, 40]
- *NCT*01688479: [0, 20]
- *NCT*01698281: [0, 40]
- *NCT*01716468: [30, 40]
- *NCT*01771666: [0, 0]
- *NCT*01805908: [0, 40]
- *NCT*01814865: [30, 40]
- *NCT*01823991: [0, 0]
- *NCT*01861509: [0, 40]
- *NCT*01866670: [0, 20]
- *NCT*01876238: [0, 20]
- *NCT*01895491: [0, 40]
- *NCT*01912963: [0, 0]
- *NCT*01927081: [0, 50]
- *NCT*01937507: [0, 20]
- *NCT*01948128: [0, 40]
- *NCT*01957514: [0, 0]
- *NCT*02066532: [0, 0]
- *NCT*02089100: [30, 40]
- *NCT*02094742: [0, 40]
- *NCT*02097238: [0, 0]
- *NCT*02185352: [30, 30]
- *NCT*02203526: [50, 60]
- *NCT*02236000: [0, 60]
- *NCT*02276443: [0, 40]
- *NCT*02288169: [30, 40]
- *NCT*02324088: [10, 40]
- *NCT*02393794: [0, 60]
- *NCT*02424682: [50, 60]
- *NCT*02443467: [50, 60]
- *NCT*02453620: [0, 0]
- *NCT*02487979: [0, 0]
- *NCT*02496065: [0, 50]
- *NCT*02514083: [10, 20]
- *NCT*02524951: [0, 40]
- *NCT*02576665: [0, 0]
- *NCT*02587689: [0, 0]
- *NCT*02596373: [0, 20]
- *NCT*02619929: [0, 40]
- *NCT*02626039: [10, 40]

- *NCT*02636582: $[0, 0]$
- *NCT*02645175: $[0, 0]$
- *NCT*02708511: $[0, 40]$
- *NCT*02721147: $[0, 40]$
- *NCT*02751710: $[30, 40]$
- *NCT*02768415: $[0, 40]$
- *NCT*02839954: $[0, 0]$
- *NCT*02850419: $[0, 40]$
- *NCT*02904135: $[10, 20]$
- *NCT*02947061: $[0, 20]$
- *NCT*02962947: $[30, 40]$
- *NCT*03032406: $[50, 60]$
- *NCT*03099200: $[0, 40]$
- *NCT*03265379: $[0, 40]$
- *NCT*03282825: $[10, 20]$
- *NCT*00002465: $[30, 40]$
- *NCT*00002529: $[0, 0]$
- *NCT*00002646: $[0, 20]$
- *NCT*00003012: $[0, 40]$
- *NCT*00003098: $[0, 0]$
- *NCT*00003418: $[0, 40]$
- *NCT*00014391: $[10, 20]$
- *NCT*00179348: $[0, 40]$
- *NCT*00217399: $[30, 40]$
- *NCT*00679783: $[0, 40]$
- *NCT*00841399: $[30, 40]$
- *NCT*00854789: $[30, 40]$
- *NCT*00929591: $[0, 40]$
- *NCT*01030250: $[10, 20]$
- *NCT*01277562: $[10, 20]$
- *NCT*01287624: $[0, 60]$
- *NCT*01387295: $[0, 40]$
- *NCT*01387373: $[0, 40]$
- *NCT*01535053: $[10, 40]$
- *NCT*01846650: $[0, 20]$
- *NCT*02348281: $[0, 40]$
- *NCT*02607215: $[0, 40]$
- *NCT*00124111: $[0, 60]$
- *NCT*00338728: $[0, 60]$
- *NCT*01835158: $[0, 70]$
- *NCT*02379247: $[0, 60]$
- *NCT*02416427: $[0, 60]$
- *NCT*02575781: $[0, 60]$
- *NCT*02774681: $[0, 60]$
- *NCT*02789657: $[0, 60]$
- *NCT*02867423: $[0, 60]$
- *NCT*01971515: $[0, 60]$
- *NCT*02341911: $[0, 60]$
- *NCT*02546934: $[0, 60]$
- *NCT*00114816: $[0, 60]$
- *NCT*00166543: $[0, 60]$
- *NCT*00176046: $[0, 50]$
- *NCT*00536081: $[0, 60]$
- *NCT*00593697: $[0, 60]$
- *NCT*00617942: $[0, 60]$
- *NCT*00820924: $[0, 60]$
- *NCT*00916877: $[0, 60]$
- *NCT*01110291: $[0, 60]$
- *NCT*01196455: $[0, 60]$
- *NCT*01240421: $[0, 60]$
- *NCT*01423695: $[0, 60]$
- *NCT*01686737: $[0, 60]$
- *NCT*01913067: $[0, 60]$
- *NCT*02244580: $[0, 60]$

- *NCT*02437318: [0, 60]
- *NCT*02521441: [0, 60]
- *NCT*02522234: [0, 60]
- *NCT*02625441: [0, 60]
- *NCT*02783794: [0, 50]
- *NCT*00331097: [0, 60]
- *NCT*00516724: [0, 40]
- *NCT*02087592: [0, 50]
- *NCT*01368107: [30, 60]
- *NCT*02892734: [30, 60]
- *NCT*00325416: [30, 60]
- *NCT*01781468: [30, 60]
- *NCT*00005926: [50, 60]
- *NCT*00076609: [90, 100]
- *NCT*00140140: [50, 60]
- *NCT*00196859: [50, 60]
- *NCT*00256217: [50, 60]
- *NCT*00721630: [50, 60]
- *NCT*00944047: [50, 60]
- *NCT*01743560: [50, 60]
- *NCT*01823835: [50, 60]
- *NCT*02139358: [50, 60]
- *NCT*00491816: [50, 60]
- *NCT*00001302: [70, 70]
- *NCT*00003855: [50, 60]
- *NCT*00083304: [70, 70]
- *NCT*00106145: [50, 60]
- *NCT*00189644: [50, 60]
- *NCT*00194727: [0, 60]
- *NCT*00194766: [50, 60]
- *NCT*00270569: [70, 70]
- *NCT*00508443: [40, 40]
- *NCT*00519168: [70, 70]
- *NCT*00526617: [50, 60]
- *NCT*00581529: [70, 70]
- *NCT*00588640: [80, 80]
- *NCT*00603408: [0, 70]
- *NCT*00617968: [0, 60]
- *NCT*00636558: [0, 60]
- *NCT*00646633: [70, 70]
- *NCT*00779285: [60, 60]
- *NCT*00896324: [70, 70]
- *NCT*00934401: [60, 60]
- *NCT*01091168: [70, 70]
- *NCT*01095003: [70, 70]
- *NCT*01127074: [80, 80]
- *NCT*01339780: [70, 70]
- *NCT*01343459: [0, 60]
- *NCT*01433562: [80, 80]
- *NCT*01555944: [50, 50]
- *NCT*01777061: [0, 60]
- *NCT*01925651: [0, 60]
- *NCT*01929941: [0, 60]
- *NCT*01942980: [70, 70]
- *NCT*01953003: [70, 70]
- *NCT*01969448: [80, 80]
- *NCT*02050620: [0, 60]
- *NCT*02102568: [50, 60]
- *NCT*02159157: [80, 80]
- *NCT*02491892: [80, 80]
- *NCT*02581670: [0, 60]
- *NCT*02585362: [0, 60]
- *NCT*02738970: [50, 60]
- *NCT*02874430: [90, 100]

- *NCT*02984683: $[0, 60]$
- *NCT*00002953: $[0, 60]$
- *NCT*00206440: $[50, 60]$
- *NCT*02866591: $[0, 60]$
- *NCT*02516540: $[0, 50]$
- *NCT*00123877: $[70, 70]$
- *NCT*00148876: $[60, 60]$
- *NCT*00155259: $[70, 70]$
- *NCT*00196820: $[60, 60]$
- *NCT*00226928: $[70, 70]$
- *NCT*00393783: $[80, 80]$
- *NCT*00515411: $[70, 70]$
- *NCT*00567554: $[80, 80]$
- *NCT*00929214: $[70, 70]$
- *NCT*00930475: $[60, 60]$
- *NCT*01172223: $[80, 80]$
- *NCT*01237457: $[60, 60]$
- *NCT*01598454: $[50, 50]$
- *NCT*02481128: $[70, 70]$
- *NCT*02617043: $[70, 70]$
- *NCT*03075072: $[70, 70]$
- *NCT*03110445: $[70, 70]$

Bibliography

[1] Mikolov et al. *Learning 200-dimensional vectors applying the skip-gram model with a window size of 5*. http://bio.nlplab.org. 2013.

[2] Millian et al. *Eligibility Criteria Text Extraction*. 2013.

[3] Musen et al. *NCBO annotator*. 2008.

[4] Pyysalo et al. *First set of medical language resources created from analysis of the entire available biomedical literature*. 2013.

[5] *An Introduction to Latent Semantic Analysis*. http://lsa.colorado.edu/papers/dp1.LSAintro.pdf. Updated: 2018-03-01.

[6] Benik J. Raschid L. Vidal. ME. Anderson P. Thor A. *PAnG: finding patterns in annotation graphs*. 2012.

[7] Aronson. *MetaMap: A Tool For Recognizing UMLS Concepts in Text*. https://metamap.nlm.nih.gov/. 2001.

[8] Raschid L. Thor A. Vidal M. Benik J. Palma G. *Mining Patterns from Clinical Trial Annotated Datasets by Exploiting the NCI Thesaurus*. Boston, USA, 2012.

[9] *Breast Cancer Clinical Trials*. https://www.breastcancertrials.org/BCTIncludes/WhyParticipate/TreatmentTrials/index.html. Updated: 2017-09-25.

[10] Brusic V. Cao X. Maloney K. "Data mining of cancer vaccine trials, a birds eye view". In: *Immunome Research 2008* 4.7 (2008).

[11] *Clinical Trials XML Data Finder*. https://clinicaltrials.gov. Updated: 2018-02-17.

[12] Vivaldi J. Cotik V. Rodriguez H. *Semantic tagging of French medical entities using distant learning*. 2015.

[13] McDonald CJ. Demner-Fushman D. Chapman WW. "What can natural language processing do for clinical decision support?" In: *Journal of Biomedical Informatics* 5.42 (2009).

[14] *ECOG Performance Status Specifications*. http://ecog-acrin.org/resources/ecog-performance-status. Updated: 2017-09-25.

[15] Adam G. Dunn, Enrico Coiera, and Florence Bourgeois. "Unreported links between trial registrations and published articles were identified using document similarity measures in a cross-sectional analysis of ClinicalTrials.gov". In: *Journal of Clinical Epidemiology* 95 (2018).

[16] Harabagiu SM. Goodwin TR. "Medical Question Answering for Clinical Decision Support". In: *Processing ACM Interantional Conference Information Knowledge Management* 1.1 (2016), pp. 297–306.

[17] Wold H. "Path models with latent variables: The NIPALS approach". In: *Quantitative sociology: International perspectives on mathematical and statistical modeling* 1.1 (1975), 307–357.

[18] National Institutes of Health. *BioPortal Ontology*. https://bioportal.bioontology.org/ontologies. 2011.

[19] Isbell L. Holmes M Gray A. "Fast SVD for large-scale matrices". In: (2007).

[20] Somvanshi P. Grover A. Pai S. Sunil S. Jain J. Kumari A. "In silico analysis of natural compounds targeting structural and nonstructural proteins of chikungunya virus". In: *F1000Research* 1.1 (2017).

[21] Donyina K. *LinkedCT*. http://linkedct.org. 2011.

[22] Burchenal J. Karnofsky D. "The clinical evaluation of chemotherapeutic agents in cancer." In: (1949). Ed. by Columbia University Press, pp. 191–205.

[23] de Bruijn B. Carini S. Martin J. Sim I. Kiritchenko S. "ExaCT: automatic extraction of clinical trial characteristics from journal publications". In: *BMC Medical Informatics and Decision Making* 56.10 (2010).

[24] U.S. Medical National Library. *PubMed*. http://www.ncbi.nlm.nih.gov/pubmed/. 1996.

[25] *Matrix Approach to Linear Regression*. http://www.stat.columbia.edu/~fwood/Teaching/w4315/Fall2009/lecture_11. Updated: 2009-10-01.

[26] *MedBravo Programming Interview Task*. https://stackoverflow.com/jobs. Published: 2015-05-15.

[27] Vorobkalov P. Melnikov M. *Metrics in Ontologies in the Medical Domain*. 2014.

[28] *NIH Clinical Trials*. https://www.nhlbi.nih.gov/studies/clinicaltrials. Updated: 2017-09-25.

[29] Manning C. Pennington J. Socher R. *GloVe: Global Vectors for Word Representation*. 2014.

[30] Hofer S. Peus D. Newcomb N. "Appraisal of the Karnofsky Performance Status and proposal of a simple algorithmic system for its evaluation". In: *BMC Medical Informatics and Decision Making* 13.1 (2013).

[31] *Protocol Registration Data Element Definitions for Interventional and Observational Studies*. http://prsinfo.clinicaltrials.gov/definitions.html. Updated: 2017-06-01.

[32] *R2 is rescaled mean squared errorr*. http://brenocon.com/rsquared_is_mse_rescaled.pdf. Updated: 2009-09-03.

[33] Sebastian Raschka. *Python Machine Learning*. Packt Publishing, 2015. ISBN: 1783555130, 9781783555130.

[34] Williams R. Ruineihart D. Hint. G. "Learning Internal Representations By Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* 1.1 (1985), pp. 1–33.

[35] Ogren P. Zheng J. Sohn S. Kipper-Schuler K. Chute C. Savova G. Masanz J. "Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications". In: *Journal of the American Medical Informatics Association* 17.5 (2010), 507–513.

[36] *Singular Value Decomposition*. http://web.cs.iastate.edu/~cs577/handouts/svd.pdf. Updated: 2017-09-12.

[37] Riehl V. *Concepts, Descriptions, and Relationships integrated into UMLS*. https://www.snomed.org/. 2014.

[38] Rodríguez H. Vivaldi J. "Using Wikipedia for term extraction in the biomedical domain: first experience". In: *In Procesamiento del Lenguaje Natural 45* 1.1 (2011), pp. 251–254.

[39] Xia F. Payne T. Yetisgen M. Gunn M. "A text processing pipeline to extract recommendations from radiology reports". In: *Journal of Biomedical Informatics* 46.2 (2013), pp. 354–362.

[40] Sonntag D. Zillner S. "Aligning medical ontologies by axiomatic models, corpus linguistic syntactic rules and context information". In: *Computer-Based Medical Systems (CBMS)* 1.1 (2011).

[41] Charles G. et al. Zubrod. "Appraisal of methods for the study of chemotherapy of cancer in man: Comparative therapeutic trial of nitrogen mustard and triethylene thiophosphoramide". In: *Journal of Clinical Epidemiology* 11.1 (1960), pp. 7–33.